

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

КУРСОВАЯ РАБОТА

по _____ Разработка кроссплатформенных приложений _____
Тема _____ Приложение «АРМ администратора ломбарда» _____

(оценка)

Руководитель

Кульков Я.Ю. _____
(фамилия, инициалы)

(подпись) _____ (дата)

Члены комиссии

(подпись) _____ (Ф.И.О.)

(подпись) _____ (Ф.И.О.)

Студент ПИН-119 _____
(группа)

Львова К.Ю. _____
(фамилия, инициалы)

(подпись) _____ (дата)

В данной курсовой работе разработано АРМ администратора ломбарда. В ходе выполнения курсовой работы произведен анализ и сбор требований к проектируемой системе. На этапе разработки структуры классов выделены основные сущности, их атрибуты и взаимосвязи. Система создавалась на языке программирования Java в среде разработки IntelliJ IDEA. Для хранения данных о клиентах, товарах и их дефектах использовалась такая база данных как MySQL. На заключительном этапе работы произведено тестирование разработанного продукта.

In this course work, the workstation of the pawnshop administrator has been developed. In the course of the course work, the analysis and collection of requirements for the designed system was made. At the stage of class development, the main entities, their attributes and relationships are highlighted. The system was created in the Java programming language in the IntelliJ IDEA development environment. For such storage of data about customers and their defects, a database like MySQL was used. At the final stage of work, the developed product was tested.

Содержание

Введение	6
1. Анализ технического задания	7
1.1 Описание предметной области	7
1.2 Анализ требований	9
2. Разработка алгоритмов	12
3 Руководство программиста	17
4. Руководство пользователя	24
5 Тестирование	35
Заключение	39
Список использованных источников	40
Приложение 1. Функциональная модель	41
Приложение 2. Скриншоты программы	44
Приложение 3 Код программы	47

					МИВУ 09.03.04-12.000 ПЗ			
Изм.	Лист	Недокум.	Подп.	Дата	Приложение «АРМ администратора ломбарда»	Лит.	Лист	Листов
Разраб.		Львова К.Ю.				у		
Пров.		Кульков Я.Ю.					5	47
						МИ ВлГУ ПИН-119		
Н.контр...								
УТВ.								

Введение

В наше время информационные технологии дают возможность рационально управлять всеми видами ресурсов предприятия, что облегчает работу предпринимателя - одиночки. Поэтому предоставляемые посредством информационных технологий сведения позволяют осуществить концентрацию ресурсов в нужное время и нужном месте для решения главных задач. Организованные в систему знания повышают компетентность сотрудников и позволяют предприятию работать рациональнее, целенаправленно и экономно, более эффективно.

В последнее время для учета операций, проводимых в ломбардах, все чаще используют информационные системы (ИС). Разработанные системы используются одним или несколькими сотрудниками, отвечающими за ведение учета операций по приему и выдачи залогов в ломбарде, выдаче кредитов, начислению процентов, сбору итоговой информации, и формированию отчетов.

Целью работы является разработка автономного кроссплатформенного приложения «АРМ администратора ломбарда».

Задачи работы:

- создать базу данных клиентов и принятого у них товара;
- организовать систему вычисления сроков выкупа;
- предусмотреть возможность просмотра информации о товарах для дальнейшего формирования отчетов.

Анализ технического задания

1.1 Описание предметной области

Ломбард - специализированная коммерческая организация, основным видом деятельности которой является предоставление коротко срочных займов под задаток движимого имущества граждан РФ. В зависимости от специализации, в ломбардах принимают драгоценный сплав, бытовые вещи, автомобильный транспорт. В курсовой работе будет рассматриваться ломбард принимает любое ценное имущество, кроме недвижимости - от драгоценностей до автомобиля.

У всякого покупателя, который хочет получить займ под залог собственного имущества, запрашиваются паспортные данные (фамилия, имя, отчество, серию и номер паспорта и т.д.)

Продавец принимает товар у клиентов и оценивает его. В течении 30 дней товар может быть выкуплен только его владельцем (если клиент постоянный этот срок может увеличиваться). По истечению срока любой другой клиент может выкупить товар. Продажная стоимость тогда будет на 30% выше закупочной.

Менеджер имеет возможность просматривать данные о товарах и клиентах.

Так же в ломбарде формируют отчет по тому залоговому имуществу, у которого заканчивается срок хранения с выводом информации о владельце.

Для того что бы понять, что должно быть в разрабатываемой системе, нам необходимо просмотреть аналоги:

Основные достоинства программы «1С Ломбард» (см. Рис.1):

Удобная и интуитивно понятная панель по каждой операции

Моментальное открытие счета на основании данных удостоверяющего документа

Подготовка и выгрузка отчетности в новом формате — XBRL

Полный бизнес-функционал:

- Ведение характеристик залогового имущества
- Проведение оценки имущества

					МИВУ.09.03.04-12.000 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подп.	Дата		

- Учёт операций по предоставлению займов под залог
- Учет операций по перезалогу имущества (изменение условий залога)
- Учет досрочного выкупа
- Реализация не востребованного имущества
- Учет операций по хранению имущества
- Инвентаризация залогового имущества
- Управленческая отчетность и т.д. [1]

← → ★ Выдача займа, получение залога: ЛУ00-000073 от 01.09.2019 12:00:01

Провести и закрыть Записать Провести Печать Создать на основании Бланк испорчен Анкета клиента (досье) Еще ?

Номер: ЛУ00-000073 от: 01.09.2019 12:00:01

Залоговый билет: Залоговый билет Подбор БСО УУУ № 00008 Организация: Ломбард "УСПЕХ"

Заемщик: Задорнов Михаил Александрович [Данные клиента](#) Склад: Склад (Залоговый билет)

Тип изделий: Ювелирное изделие Подразделение: головное

Форма оплаты: Наличные По карте

Срок (дней): 7 Дата окончания: 07.09.2019 Лыготный период до: 07.10.2019

Условия договора: Экспресс

ПСК: 109,500 % 10 210,00 руб.

Предметы залога (1)

Добавить Скопировать Оценка стоимости залога, займа Еще -

N	Номенклатура	Ед.	Количество	Оценка за ед.	Оценка предмета	Сумма займа	Счет учета
1	Кольцо Б/У Зол 585 о.в. 10 гр.	шт	1,000	13 580,00	13 580,00	10 000,00	008.01

Условия договора: ЛУ00-000073 от 01.09.2019

Еще ?

Оценка предметов: 13 580,00 Сумма займа: 10 000,00 На срок: 7 Дата окончания: 07.09.2019 Лыготный период до: 07.10.2019

Условия договора: Экспресс

Расчет условий договора

N	Вид начисления	Формула расчета	Ставка	Вид ставки	Количество	Расчетная сумма	Дата начала действия	Дата окончания по догово
1	Процент за пользо	% от займа в день	0,30	%	7,000	210,00	01.09.2019	07.09.2019

Расчет начислен

Процент за пользо

Рис. 1 - Программа «1С Ломбарда»

Основные достоинства программы «Мой ломбард» (см. Рис.2):

Удобный учет залоговых операций

Полная реализация отчетности

Система учитывает все нововведения в учете займов и залогов, таких как расчет эффективной ставки процента по займу, расчет амортизированной стоимости займа и отражение корректировок по первоначальному признанию [2].

Движения документа: Поступление наличных 245-1 от 26.04.2018 11:57:20				
Занесать и закрыть Обновить Открыть значение <input type="checkbox"/> Ручная корректировка (разрешает редактирование движений документа) Бухгалтерский и налоговый учет (3) ЕПС (5)				
N	Дата	Дебет	Кредит	Сумма
1	26.04.2018 11:57:20	488.02 4880261000000000769 Сделка 26/000002 от 20.04.2018 9:02:25	710.01 7100161000311800001 Хабаровск 76.06(ДТ) -> 488.02 90.01.1(КТ) -> 710.01КОР.СЧ.:76.06	400.00 Начисление процентов по договору займа №26/000002 от 20.04.2018 0:00:00
2	26.04.2018 11:57:20	202.02 20202000000022221 Касса Хабаровск	488.09 4880961000000000639 Сделка 26/000002 от 20.04.201...	400.00 Приняли проценты по договору займа №26/000002 от 20.04.2018 0:00:00 50.01(ДТ) -> 202.02 76.06(КТ) -> 488.09
3	26.04.2018 11:57:20	202.02 20202000000022221 Касса Хабаровск	612.17 6121761000000000057 Сделка 26/000002 от 20.04.201...	5 000.00 Возврат основного долга по договору займа №26/000002 от 20.04.2018. Досрочное пог... 50.01(ДТ) -> 202.02
4	26.04.2018 11:57:20	612.17 6121761000000000057 Сделка 26/000002 от 20.04.2018 9:02:25	488.01 48801610000000001793 Сделка 26/000002 от 20.04.201...	5 000.00 Возврат основного долга по договору займа №26/000002 от 20.04.2018. Досрочное пог... 58.03(КТ) -> 488.01
5	26.04.2018 11:57:20	488.09 4880961000000000639 Сделка 26/000002 от 20.04.2018 9:02:25	488.02 4880261000000000769 Сделка 26/000002 от 20.04.201...	400.00 Закрытие оплаты на начисление текущего месяца.

Рис. 2 - Программа "Мой ломбард"

1.2 Анализ требований

Для разработки приложение необходимо использовать информацию, полученную в ходе изучения аналогов и требований из технического задания. В данной курсовой работе необходимо реализовать:

Базу клиентов и принятого у них товара;

Изменение статуса товара (в течении срока хранения - 30 дней только хозяин может выкупить товар, после этого срока любой клиент может выкупить товар);

Изменения срока гарантийного хранения на усмотрение работника ломбарда (если клиент постоянный, то срок может быть увеличен);

Возможность просмотра:

- сколько и на какую сумму было закуплено товаров у определенного поставщика за указанный период времени
- на какую сумму было принято товаров
- на какую сумму было продано указанных товаров

Вывод информации по тому залоговому имуществу, у которого заканчивается срок гарантийного хранения с выводом информации о владельцах.

Проанализировав требования можно составить диаграмму прецедентов (см. Рис.3), показывающая общий функционал разрабатываемой программы.

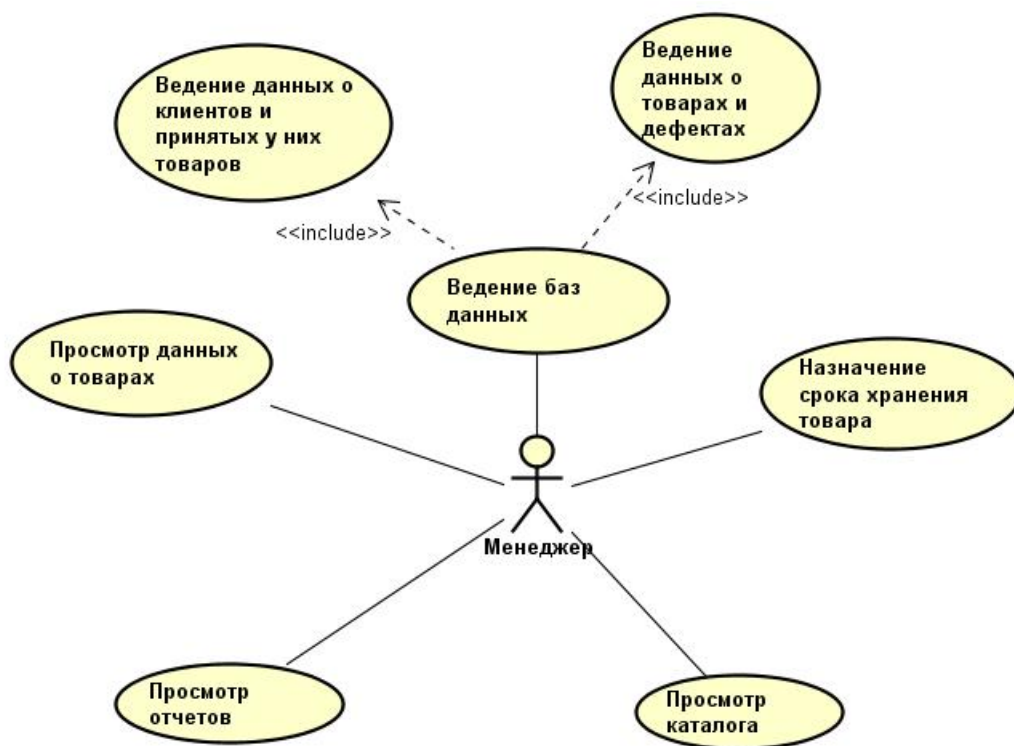


Рис. 3 - Диаграмма прецедентов

АРМ предназначена для одного пользователя: Администратора. Он может добавлять товары и клиентов в соответствующие базы данных, дополнять, изменять и удалять данные о товарах. дефектах и клиентах, назначать индивидуальные для каждого клиента сроки хранения товара. Система должна формировать отчеты и каталог товаров.

Чтобы разработать АРМ ломбарда выбрана СУБД MySQL, т.к. она имеет такие достоинства [3] как:

MySQL является наиболее безопас-ной и надежной системой управления базами данных, используемой многи-ми известными предприятиями, такими как Facebook, Twitter и Wikipedia..

MySQL - это система управления базами данных, наилучшим образом приспособленная для масштабируемости, которая может масштабироваться горизонтально и вертикально; с точки зрения данных и нагрузки.

MySQL - это реляционная система управления базами данных с открытым исходным кодом, которая делает отладку, обновление и расширение функциональности быстрым и легким.

СУБД MySQL обеспечивает обработку высокоскоростных транзакций с оптимальной скоростью.

СУБД MySQL обеспечивает кросс-платформенную гибкость, которая может работать на различных платформах, таких как Windows, Linux, Solaris, OS 2 и т. д.

В качестве среды для разработки прикладной программы для работы с созданной в MySQL базой данных, была выбрана интегрированная среда IntelliJ IDEA, язык программирования Java. IntelliJ IDEA обеспечивает быстрый доступ ко всем функциям и встроенным инструментам, необходимым разработчику, а также широкие возможности индивидуальной настройки.

2. Разработка алгоритмов

Для разработки точных алгоритмов, включающих в себя всю логику разрабатываемой программы, необходимо провести моделирование данных используемых при разработке приложения.

Моделирование данных — это создание визуального представления о всей информационной системе либо ее части. Цель в том, чтобы проиллюстрировать типы данных, которые используются и хранятся в системе, отношения между этими типами данных, способы группировки и организации данных, их форматы и атрибуты [2].

Концептуальная модель — это модель, представленная множеством понятий и связей между ними, определяющих смысловую структуру рассматриваемой предметной области или её конкретного объекта [4].

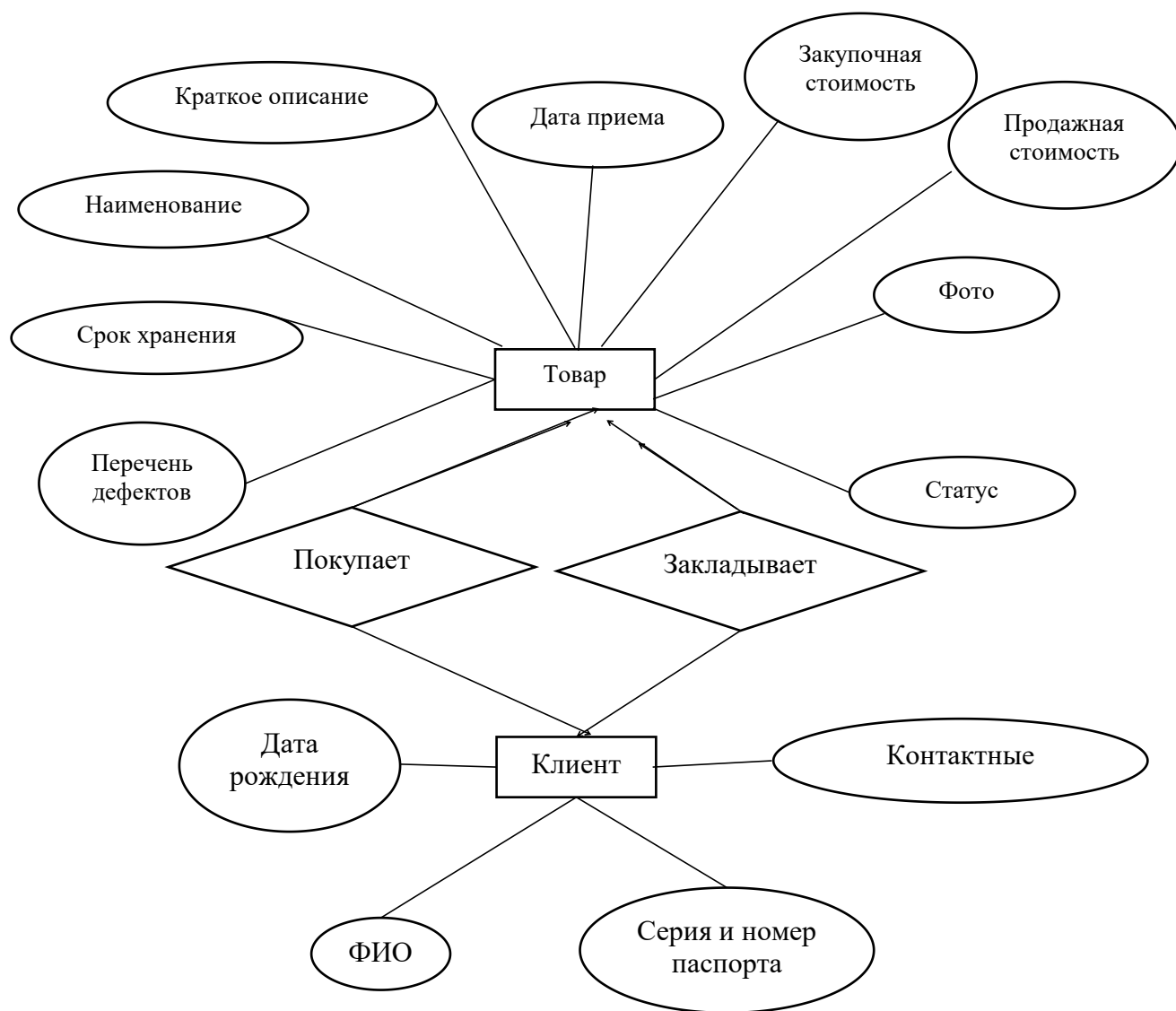


Рис. 4 - Концептуальная модель

Логическая модель – графическое представление структуры базы данных с учетом принимаемой модели данных (иерархической, сетевой, реляционной и т.д.), независимое от конечной реализации базы данных и аппаратной платформы.

Иными словами, она показывает, ЧТО хранится в базе данных (объекты предметной области, их атрибуты и связи между ними), но не отвечает на вопрос КАК (рис. 5) [5].

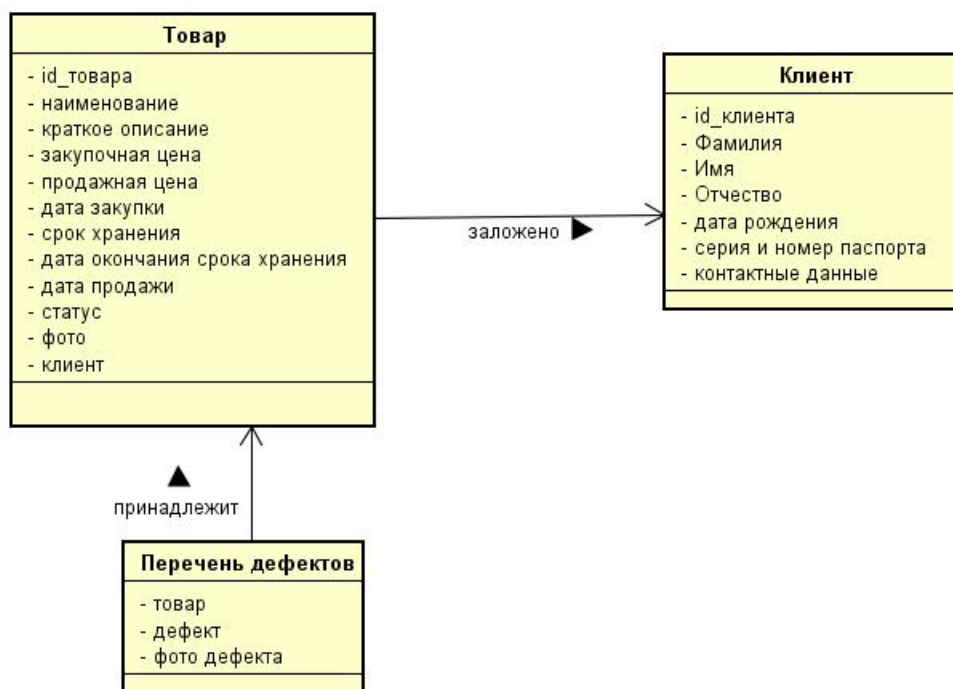


Рис. 5 - Логическая модель

Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей и т.п. (рис. 6).

Физическая модель строится на основе логической с учетом ограничений, накладываемых возможностями выбранной СУБД (в данном случае - MySQL).

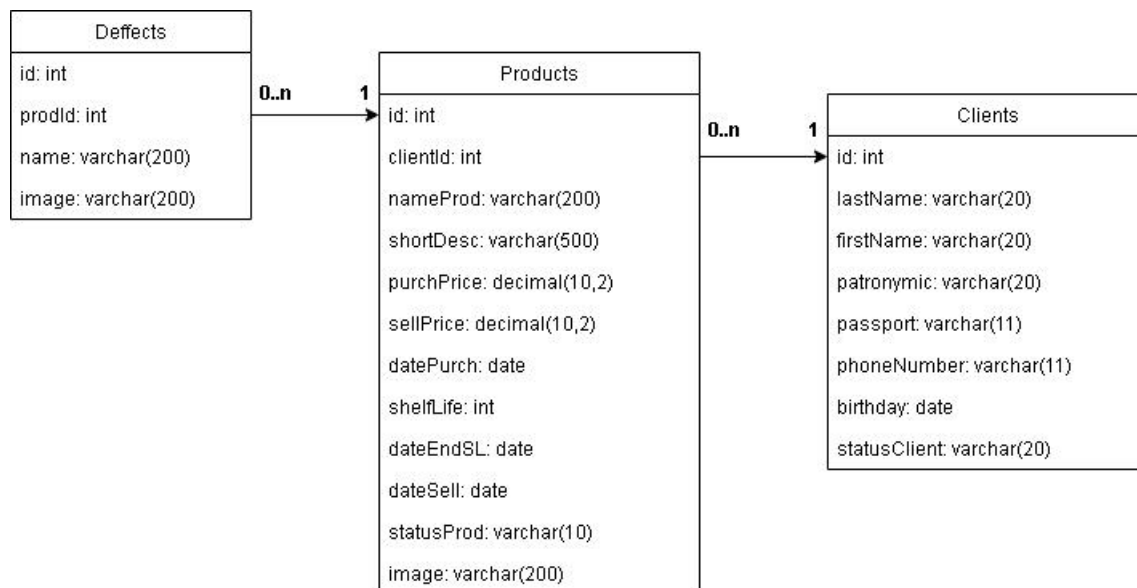


Рис. 6 - Физическая модель

После была создана физическая модель, для явного представления таблиц в базе данных.

Таблица Clients хранит в себе данные о клиентах, которые заложили товар/товары.

Поле id – идентификатор клиента.

Поле firstName – имя клиента.

Поле lastName – фамилия клиента.

Поле patronymic – отчество клиента.

Поле birthday– дата рождения, для проверки совершеннолетия клиента.

Поле phoneNumber – контактный номер телефона.

Поле passport – серия и номер паспорта, для проверки личности клиента.

Таблица Products (см. Приложение 1) хранит данные о товарах.

Поле id – идентификатор товара.

Поле name – наименование товара.

Поле shortDesc– краткое описание товара.

Поле purchPrice – цена при заложении товара.

Поле sellPrice– цена для продажи товара (на 30% больше цены при заложении).

Поле datePurch – дата заложения товара.

Поле shelfLife – срок хранения товара.

Поле dateEndSL – дата окончания срока хранения товара.

Поле dateSall – дата продажи товара.

Поле statusProd – статус состояния товара
(заложено/продается/продано/выкуплено).

Поле image – имя фотографии.

Поле ClientId – идентификатор клиента, которому принадлежит данный товар.

Таблица Defects хранит данные о дефектах товаров.

Поле id – идентификатор дефекта.

Поле name – наименование дефекта (краткое описание).

Поле image – имя фотографии дефекта.

Поле prodId – идентификатор товара, которому принадлежит данный дефект.

Функциональная модель, аналогичная модели деятельности или модели процесса, представляет собой графическое представление функции предприятия в определенной области. Цели функциональной модели заключаются в описании функций и процессов, оказании помощи в выявлении информационных потребностей, определении возможностей и создании основы для определения затрат на продукцию и услуги [6].

Функциональная модель представляет такие функции и алгоритмы разрабатываемой системы, как добавление, изменение, удаление данных о клиентах, товарах и их дефектах (см. Приложение 1).

3 Руководство программиста

Во время разработки приложения был использован инструмент, для конструирования окон – Scene Builder.

С его помощью были разработаны fxml представления:

1) hello-view.fxml – основное окно для работы с данными клиентов, товаров и клиентов, а так же формирования каталога продаваемых товаров и отчетов.

Данное окно является основным. В нем элементы разделены на несколько “разделов”, для переключения по ним используются кнопки, соответствующие разделам:

- bCatalog – button для перехода в раздел «Каталог», который содержит:
spCatalog – splitPanel для работы с каталогом разделен на две визуальные части. Одна содержит компоненты для поиска товаров, вывода списка товаров и продажи выбранного товара.

- bProdSearchCatalog – button для поиска товара по наименованию вв веденному в textField tfSearchProdCatalog;

- tvCatalogTable – tableView выводит в tableColumn tcNameProdCatalog наименования товаров;

- dBuyProd – button для продажи товаров;

Вторая часть панели содержит поля для вывода информации и выбранном товаре из таблицы.

- lClientProdCatalog – lable для вывода данных о клиенте, заложившего товар;

- lNameProdCatalog – lable для вывода наименования товара;

- taDescProdCatalog – textArea для вывода краткого описания товара;

- lSellPriceProdCatalog - lable для вывода цены продажи;

- ivImgProdCatalog – imageView для вывода фотографии товара.

- bClients – button для перехода в раздел «Клиенты», который содержит:
vbClients – VBox для вывода информации о клиенте для редактирования
включает такие элементы как:

- lStatusClient – label для вывода статуса клиента.
- tfLastname – textField для вывода фамилии клиента.
- tfFirstname – textField для вывода имени.
- tfPatronymic – textField для вывода отчества.
- dpBirthday – datePicker для вывода даты рождения.
- tfPassport – textField для вывода данных паспорта.
- tfPhoneNumber – textField для вывода номера телефона.
- bEditClient – button для сохранения внесенных изменений.
- bDeleteClient – button для удаления клиента.

Другим компонентом раздела является spClients – splitPanel для вывода списка клиентов и работы с их товарами. Включает в себя такие элементы как:

- bSearchClient – button для поиска клиента по данным паспорта
введенным в textField tfSearchClient.

- bAddClient – button для открытия окна для добавления клиента.
- tvTableClients – tableView для вывода списка клиентов.
- tcLastname – tableColumn для вывода фамилий клиентов.
- tcFirstname – tableColumn для вывода имен.
- tcBirthday – tableColumn для вывода даты рождения.
- tcPassport – tableColumn для вывода данных паспорта.
- tcPhoneNumber – tableColumn для вывода номера телефона.
- tcStatusClient – tableColumn для вывода статуса
(постоянный/непостоянный).

- bAddProd – button для открытия окна добавления товара, выбранного из
таблицы клиента.

- tvTableProducts – tableView для вывода товаров, выбранного клиента.
- tcName – tableColumn для вывода наименований товаров.

- tcStatusProd - tableColumn для вывода статусов товаров.
- ivImgProd – imageView для вывода фотографии, выбранного из таблицы товара.
- Кнопка «Подробнее» открывает форму для просмотра информации о товаре, выбранном из таблицы.
- bProducts – button для перехода в раздел «Товары», который содержит splitPanel spProducts включающий в себя:
 - bProdSearch – button для поиска товаров по наименованию введенного в textField tfSearchProd
 - tvProductTable - tableView для вывода товаров.
 - tcProdName - tableColumn для вывода наименований товаров.
 - tcProdStatus - tableColumn для вывода статусов товаров.
 - ivProdImg - imageView для вывода фотографии, выбранного из таблицы товара.
 - bUpdateProd – button для открытия окна для редактирования данных о товаре.
 - dDeleteProd – button для удаления товара.
 - bAddDef – button для открытия окна для добавления нового дефекта, для выбранного товара.
 - tvTableDef - tableView для вывода дефектов выбранного товара.
 - tcNameDef - tableColumn для вывода описаний дефектов
 - ivImgDefect – imageView для вывода фотографии, выбранного дефекта.
 - tfNameDef – textField для редактирования описания, выбранного дефекта.
 - bLoadImgDef - button для загрузки новой фотографии дефекта.
 - bUpdateDef - button для сохранения изменений дефекта.
 - bDeleteDef - button для удаления дефекта.
- bReports – button для перехода в раздел «Отчеты», который содержит anchorPane apReport для работы с отчетами. Данная панель содержит в себе

splitMenuButton splitMenuReport для выбора отчета, с которым будет работать пользователь. От этого выбора зависит какой splitPane станет доступным:

- spReportClientProd включает в себя такие элементы как:
 - cbReportClient – comboBox предоставляет возможность выбрать клиента для отчета.
 - cbReportTime – comboBox предоставляет возможность выбрать период времени.
 - Кнопка «Вывести товары» выводит в tableView tvProductReport список товаров по выбранным критериям.
 - tcNameProdReport1 – tableColumn для вывода наименований товаров.
 - tcPriceProdReport – tableColumn для вывода цен.
 - taInfProdReport – textArea выводит количество и сумму товаров, из таблицы.
- spReportProduct включает в себя такие элементы как:
 - tvProdEndSL – tableView для вывода информации о товарах.
 - tcNameProdReport – tableColumn для вывода наименований товаров.
 - tcDatePurch – tableColumn для вывода дат заложения.
 - tcDateEndSL – tableColumn для вывода дат окончания срока хранения.
 - taInfClientProdEndSL – textArea для вывода информации о клиенте выбранного из таблицы товаров.

Для реализации функционала используется класс HelloController.

2) addClient.fxml – окно для добавления нового клиента.

Данное представление содержит такие элементы как:

- tfLastName – textField для ввода фамилии клиента;
- tfFirstName – textField для ввода имени;
- tfPatronymic – textField для ввода отчества;
- dpBirthday – textField для ввода даты рождения;

- tfPassport – textField для ввода паспорта
- tfPhoneNumber - textField для ввода номера телефона;
- bAddClient – button для добавления нового клиента.

Для реализации функционала используется класс NewClientController.

3) addProd.fxml – окно для добавления нового товара.

Данное окно содержит такие элементы как:

- lClient – lable для вывода данных клиента, заложившего товар;
- tfNameProd – textField для ввода наименования товара;
- taShortDesc – textArea для ввода описания товара;
- spinShelfLife – spinner для ввода срока хранения, если клиент

постоянный;

- dpDatePurch – textField для вывода даты заложения (сегодняшняя дата);
- dpDateEndSL – textField для вывода даты окончания срока хранения;
- tfPricePurch - textField для ввода цены заложения;
- ivImgProd – imageView для вывода загруженной фотографии;
- bLoadImgProd – button для загрузки фотографии товара;
- bAddProduct - button для добавления нового товара;
- lErrorAddProd – lable для вывода ошибок ввода данных.

Для реализации функционала используется класс NewProductController.

4) allinfProduct.fxml – окно для просмотра информации о товаре.

Для вывода информации о товаре используются следующие элементы:

- tfNameProd – textField для вывода наименования товара;
- taShortDesc – textArea для вывода описания;
- tfDatePurch – textField для вывода даты заложения;
- tfShelfLife – textField для вывода срока гарантийного хранения;
- tfDateEndSL - textField для вывода даты окончания срока хранения;
- tfDateSell - textField для вывода даты продажи, если товар продан, иначе

поле скрыто;

- tfPurchPrice - textField для вывода цены заложения;
- tfSellPrice - textField для вывода цены продажи;

- ivImg – imageView для вывода фотографии товара;
- кнопка «Вернуться» для возвращения на основное окно.

Для реализации функционала используется класс AllInfProdController.

5) updateProduct.fxml – окно для редактирования данных о товаре.

Данное окно дает только просматривать или редактировать определенные поля в зависимости от статуса товара. Поля дат заложения, окончания срока хранения и продажи, а также поле цены продажи можно только просматривать.

Если товар «заложен», то окно позволяет редактировать только такие поля как наименование, кратное описание, цены заложения, фотографию и срок хранения (в зависимости от статуса клиента).

Если товар «продается», то разрешено редактирование наименования, описания и фотографии.

Если товар «продан» или «выкуплен», то запрещается редактирования всех полей.

Данное представление содержит такие элементы как:

- lClient – lable для вывода данных клиента заложившего данный товар;
- tfName – textField для редактирования наименования товара;
- taShortDesc – textArea для редактирования краткого описания товара;
- spShelfLife – spinner для редактирования срока хранения;
- labShelfLife – lable для вывода срока хранения, если клиент непостоянный;

- tfDatePurch – textField для вывода даты заложения товара;
- tfDateEndSL – textField для вывода даты окончания срока хранения;
- tfDateSell – textField для вывода даты продажи, если товар продан;
- tfPurchPrice – textField для редактирования цены заложения;
- tfSellPrice – textField для вывода цены продажи;
- ivImage – imageView для вывода фотографии товара;
- bLoadImg – button для загрузки фотографии товара;
- textStatus – text для вывода статуса товара;
- bUpdate – button для сохранения внесенных изменений.

Для реализации функционала используется класс UpdateProdController.

6) addDefect.fxml – окно для добавления дефекта.

Данное окно содержит такие элементы как:

- INameProd – text для вывода наименования товара, которому мы добавляем дефект;
- tfNewNameDef – textField для ввода описания дефекта;
- ivNewImgDef – imageView для вывода загруженной фотографии;
- кнопка «Загрузить новую фотографию» для загрузки изображения;
- кнопка «Добавить дефект» для добавления дефекта с введенными данными.

Для реализации функционала используется класс NewDefectController.

Окно hello-view.fxml использует класс HelloController, в котором реализована работа над данными клиентов, товаров и дефектов, а также выкуп и продажа товара. Так же в данном классе формируются материалы для отчетов о товарах и каталог продаваемых товаров.

Для корректной работы с базой данных (добавления, изменения и удаления данных из нее) по моделям данных были созданы вспомогательные классы:

- 1) ClientsClass – данные о клиенте.
- 2) Product – данные о товаре.
- 3) Deffect – данные о дефекте.
- 4) LoadImage – для выгрузки фотографий по именам взятым из базы данных и сохранение их в подкорневой каталог проекта.

Для улучшения восприятия пользовательского интерфейса была язык CSS, а именно подключена DarkTheme.css

Более подробное документирование кода представлено по ссылке:
<https://github.com/LvovaKs/Pavnshop.iml.git>.

4. Руководство пользователя

Разрабатываемое приложение должно позволять пользователю добавлять, редактировать и удалять данные о клиентах, товарах и дефектах, а так же выкупать/продавать товары и иметь возможность просматривать данные о товарах с определенными фильтрами.

Для того что бы добавить нового клиента необходимо выбрать раздел «Клиенты» в боковом меню (см. Рис. 7).

The screenshot shows a web application window titled 'Hello!'. The interface is divided into several sections:

- Sidebar Menu:** Contains links for 'Каталог', 'Клиенты' (highlighted in green), 'Товары', 'Отчеты', and 'Процесс работы: Информация о клиентах загружена'.
- Search and Add:** A search bar with a 'поиск' button and a 'Добавить' button.
- Table:** A table with columns: 'Фамилия', 'Имя', 'Отчество', 'Дата рожд...', 'Паспортн...', 'Контактны...', and 'Статус'. It contains five rows of client data.
- Form:** A form for adding or editing a client, with fields for 'Фамилия', 'Имя', 'Отчество', 'Дата рождения' (with a dropdown), 'Паспортные данные', and 'Контактные данные'. It includes 'Изменить' and 'Удалить' buttons.
- Buttons:** 'Добавить товар' and 'Подробнее' buttons are located below the table.

Фамилия	Имя	Отчество	Дата рожд...	Паспортн...	Контактны...	Статус
Петров	Иван		2001-01-04	1596 125698	89563695475	не постоян...
Симонова	Анна	Юрьевна	2000-05-25	1000 000000	890000000000	не постоян...
Осипов	Милан		2001-05-03	1000 000001	890000000001	не постоян...
Миронова	Мария		1992-05-08	1000 000003	890000000003	не постоян...

Рис. 7 - раздел «Клиенты»

Далее нажать кнопку «Добавить» рядом с поиском. Откроется окно для добавления (см. Рис. 8).

Рис. 8 – добавление клиента.

Если все данные введены верно, то клиент будет добавлен, если нет, то выведется ошибка.

Для редактирования данных о клиенте необходимо выбрать клиента из таблицы клиентов (см. рис. 9).

Рис. 9 – выбор клиента из таблицы

В боковые поля будет занесены все данные о клиенте. Если все данные внесены верно, то при нажатии кнопки «Изменить» данные будут изменены (см. Рис. 10).

поиск

не постоянный

Фамилия	Имя	Отчество	Дата рожд...	Паспортн...	Контактны...	Статус
Петров	Иван		2001-01-04	1596 125698	89563695475	не постоя...
Симонова	Анна	Юрьевна	2000-05-25	1000 000000	890000000000	не постоя...
Осипов	Милан		2001-05-03	1000 000001	890000000001	не постоя...
Миронова	Мария	Игоревна	1992-05-08	1000 000003	890000000003	не постоя...
Львова	Ксения	Юрьевна	1982-06-11	1000 000004	890000000004	не постоя...

Каталог

Клиенты

Товары

Отчеты

Процесс работы:

Информация о клиентах загружена

Добавить товар

Наименование	Статус
Отбойный молоток Makita...	заложено
Ноутбук acer aspire ES1-73...	продается
Нивелир Elitech LH 5	заложено

Подробнее

Фамилия

Имя

Отчество

Дата рождения

Паспортные данные

Контактные данные

Рис. 10 – изменение данных клиента

Для удаления выбранного клиента необходимо нажать кнопку «Удалить» (см. Рис. 11).

Фамилия	Имя	Отчество	Дата рожд...	Паспортн...	Контактны...	Статус
Петров	Иван		2001-01-04	1596 125698	89563695475	не постоя...
Симонова	Анна	Юрьевна	2000-05-25	1000 000000	890000000000	не постоя...
Осипов	Милан		2001-05-03	1000 000001	890000000001	не постоя...
Миронова	Мария	Игоревна	1992-05-08	1000 000003	890000000003	не постоя...

Рис. 11 – удаление клиента

Для добавления товара необходимо выбрать клиента, который закладывает товар, и нажать кнопку «Добавить товар». Откроется окно для добавления товара (см. Рис. 12).

Добавление нового товара

Клиент: 1000 000003 Миронова Мария Игоревна

Наименование
Радиоприемник FORZA

Дата заложения
2022-05-05

Краткое описание
аккумуляторный портативный радиоприемник
Материал: Пластик
Радиус действия: 10 м
Тип подключения: USB, Micro-SD
слот, FM 87,5-

Дата окончания срока хранения
2022-06-04

Цена заложения
690

Срок хранения
30

загрузить изображение

не постоянный

Добавить товар

Рис. 12 – добавление нового товара

Если все данные введены верно, то товар будет добавлен (см. Рис. 13).

Редактирование

Товар клиента: 1000 000003 M... пост...

Наименование: Радиоприемник FORZA 1

Срок хранения: 30

Цена заложения: 690.0

Краткое описание: Работает только от сети; Тип: аккумуляторный портативный радиоприемник; Материал: Пластик; Радиус действия: 10 м; Тип подключения: USB, Micro-SD слот, FM 87,5-

Дата заложения: 2022-05-05

Цена продажи: 897.0

Дата окончания срока хранения: 2022-06-04

Статус товара: продается

Загрузить фотограф...

Редактировать

Рис. 14 – редактирование товара

Если все данные внесены верно, то изменения сохраняются, иначе выводится ошибка (см. Рис. 15).

Наименование товара	Статус
Брендовые часы	продается
Ноутбук acer aspire ES1-732-P2VK	продается
Нивелир Elitech ЛН 5	заложено
Крест нательный, золото 585 (14К), вес 1.4...	заложено
Кольцо с камнем, золото 585 (14К), вес 5.40 г	заложено
Радиоприемник FORZA 1	продается
Утюг STERLINGG ST-1005	продается
Паровой утюг Philips Azur Supervapor 55	продается

Рис. 15 – сохранение изменений

Для удаления товара необходимо в том же разделе нажать кнопку «Удалить товар», она может быть недоступна, т. к. товар продан или выкуплен.

Для добавления дефекта нужно выбрать товариз таблицы и нажать кнопку «Добавить дефект». Откроется форма добавления дефекта (см. Рис. 16).



Рис. 16 – добавление дефекта

Если описание занесено и картинка добавлена, то дефект добавиться в таблицу (см. Рис. 17).

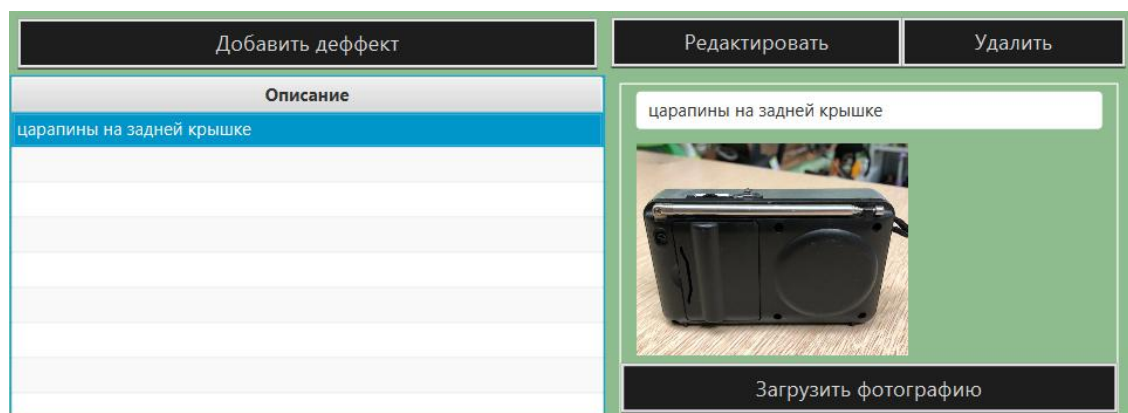


Рис. 17 – дефект добавлен

Для изменения данных дефекта нужно внести изменения в поле с описанием дефекта и/или, нажав кнопку «Загрузить фотографию», загрузить новое фото.

После нажатия кнопки «Редактировать» данные будут изменены (см. Рис. 18).



Рис. 18 – сохранение изменений

Удаление дефекта происходит по нажатию кнопки «Удалить» при выбранном дефекте из таблицы.

Для продажи товара необходимо выйти в раздел «Каталог» и выбрав товар из таблицы нажать кнопку «Продать» (см. Рис. 19-20).

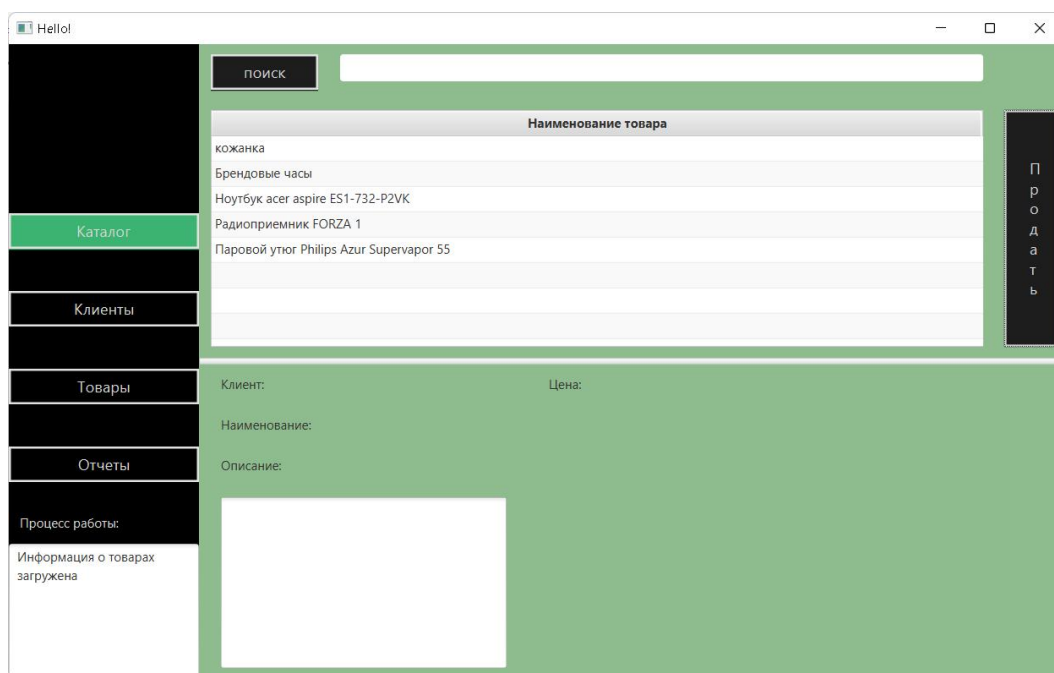


Рис. 19 – раздел «Каталог»

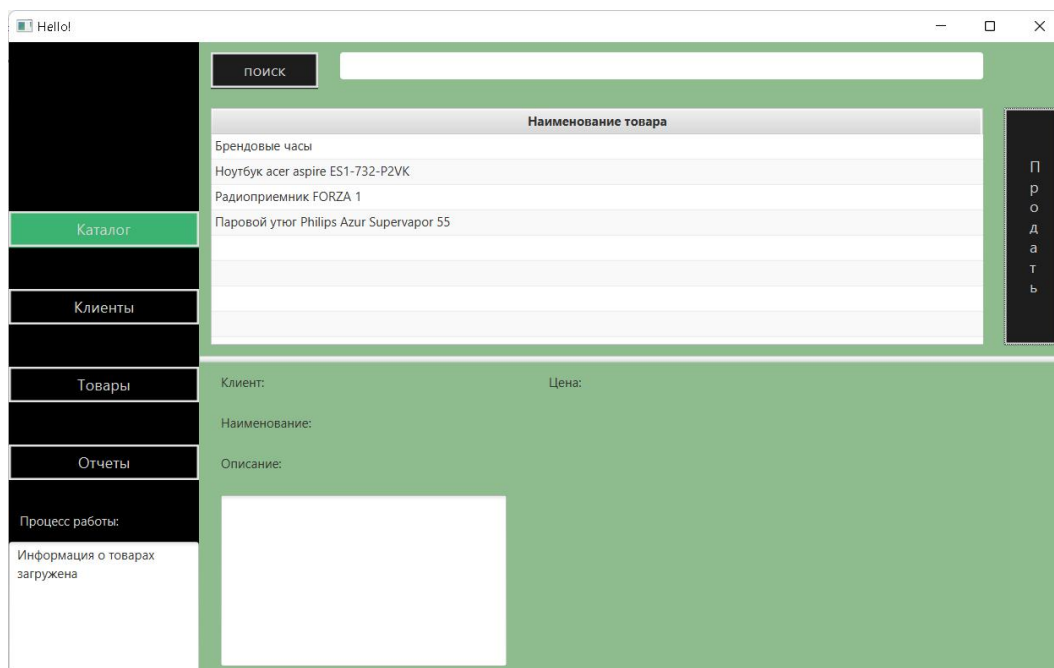


Рис. 20 – товар продан

Для просмотра отчетности необходимо выбрать раздел «Отчеты» и выбрать отчет, который хотите посмотреть (см. Рис. 21-22).

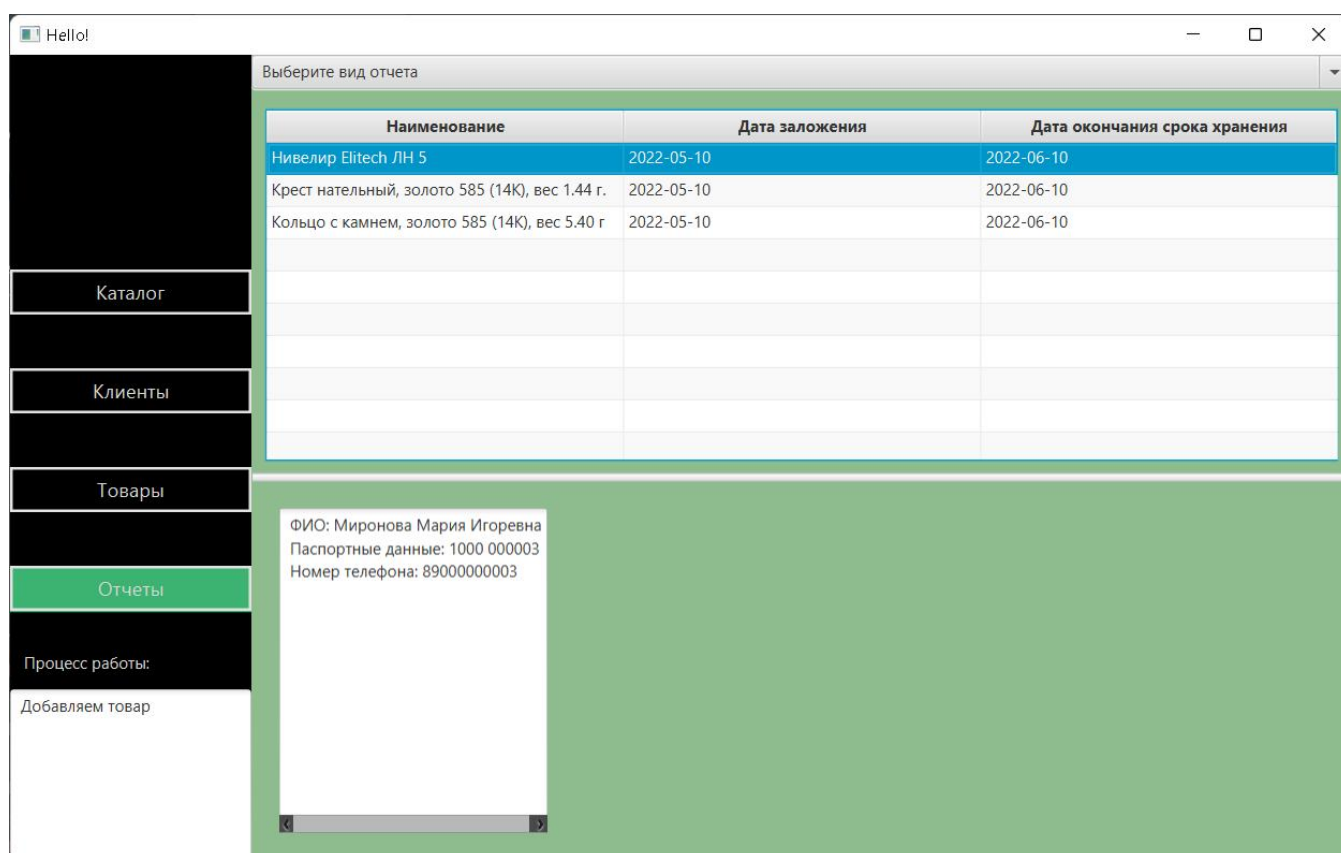


Рис. 21 – вывод товаров, у которых заканчивается срок хранения.

Рис. 22 – вывод товаров заложенных/проданных с определенными условиями

Для того что бы получить нужный отчет необходимо в выпадающих списках выбрать:

- какие товары заложенные или проданные;
- за какой промежуток времени;
- если выбраны заложенные товары, то выбрать поставщика.

Пример показан на рис. 23.

Каталог

Клиенты

Товары

Отчеты

Процесс работы:

Проверка даты

Выберите вид отчета

1000 000003 Миронова Мария Иго...

за год

заложено

Вывести товары

Отчетность:

Количество: 6

Сумма: 26840

Наименование	Цена, руб
Отбойный молоток Makita HM 1203 C	2000.0
Ноутбук acer aspire ES1-732-P2VK	20000.0
Нивелир Elitech ЛН 5	2500.0
Радиоприемник FORZA 1	690.0
Утюг STERLINGG ST-1005	250.0
Паровой утюг Philips Azur Supervapor 55	1400.0

Рис. 23 – пример отчета с заданным поставщиком и периодом времени

5 Тестирование

Тестирование – процесс многократного повторения программы с целью обнаружения ошибок.

- подключение к базе данных

При загрузке приложения производится подключение к базе данных, если подключиться не удалось, то система выведет ошибку (см. Рис. 25).



Рис. 25 – ошибка подключения к базе данных

- добавление нового клиента:

При добавлении нового клиента проходит проверка на корректный ввод фамилии, имени, отчества, номера телефона, паспортных данных, а также даты рождения (клиент должен быть совершеннолетним).

При некорректном вводе фамилии, имени и отчества выводится ошибка показанная на Рис. П2.1 и Рис. 26.

Рис. 26 – Некорректный ввод данных ФИО

Если дата введена не верно, то система выдает ошибку показанную на Рис. П2.1.

Если номер телефона содержит буквы, то система выдает ошибку – рис. 26.

Рис. 27 – Номер телефона неверен

Если пользователь ввел данные паспорта и/или номер телефона, которые принадлежат другому клиенту, то он не будет добавлен.

- Изменение данных клиента:

При изменении данных клиента проводятся те же проверки, что и при добавлении (см. Рис. П2.1 - 4).

- добавление товара

При добавлении товара проводится проверка на введенную цену заложения, если проверка не пройдена, то выводится ошибка (см. Рис. П2.5). А также идет проверка на то, чтобы все данные были введены (пример на Рис. 28 - 29).

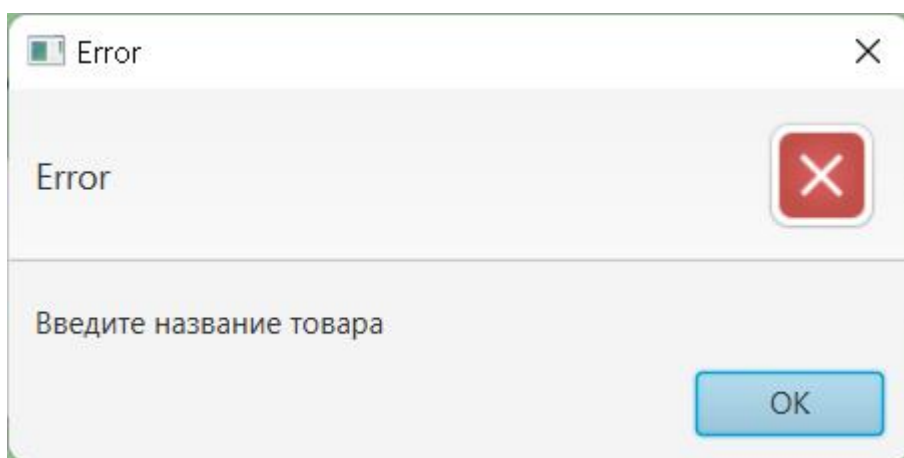


Рис. 28 – ошибка добавления товара

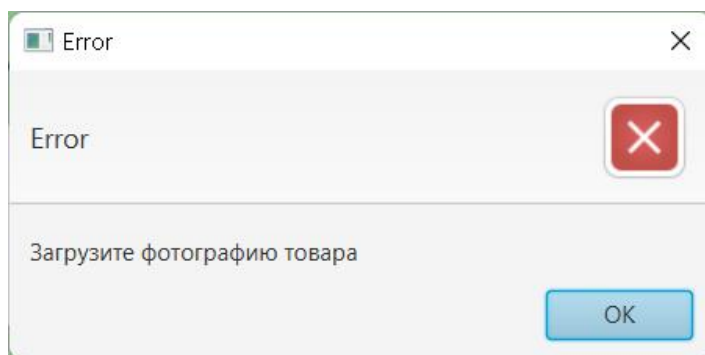


Рис. 29 – не загружена фотография товара при добавлении

- изменение данных товара

При изменении данных тавара проводятся те же проверки, что и при добавлении.

- добавление дефекта

При добавлении дефекта проводится проверка на то, что бы все данные были введены. Если хотя бы одно поле не заполнено – система выводит ошибку (см. Рис. П2.6).

- изменение дефекта

При изменении дефекта проводится проверка на ввод «описания». Если описания нет – то выводится ошибка (см. Рис.30).

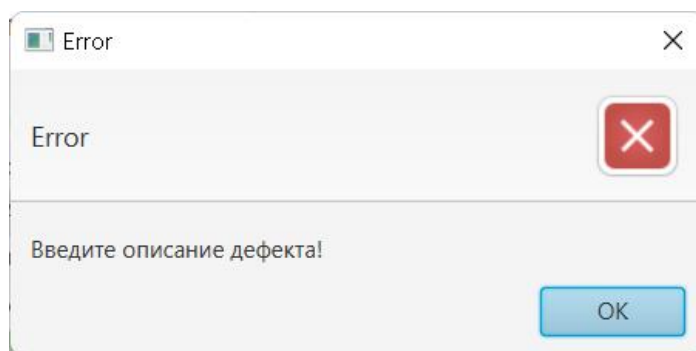


Рис. 30 – не введено описание дефекта при сохранении изменений

- составление отчета

При составлении отчета по принятым/проданным товарам проверяется ввод данных для отчета, если их не достаточно, то выводится ошибка (см. Рис. П2.7)

Заключение

В ходе курсовой работы было создано автономное кроссплатформенное приложение «АРМ администратора ломбарда» и были решены следующие задачи:

- создана база данных клиентов и принятого у них товара;
- организовать систему вычисления сроков выкупа;
- предусмотреть возможность просмотра информации о товарах для дальнейшего формирования отчетов;
- дать администратору возможность изменять сроки хранения товаров для постоянных клиентов;
- разработать приложение, отвечающее всем требованиям технического задания;
- протестировать готовое приложение.

В ходе разработки использовалась база данных MySQL и язык программирования Java.

В будущем в АРМ можно добавить возможность формировать отчеты автоматически, используя данные из БД.

Список использованных источников

1. 1СЛомбард [Электронный ресурс].URL: https://1c-lombard.ru/?utm_source=yandex&utm_medium=cpc&utm_campaign=65616492&utm_content=campaign:typel|gbid:4691401783|aid:11156080009&utm_term=программа%201c%20ломбард|&calltouch_tm=yd_c:65616492_gb:4691401783_ad:11156080009_ph:33989150542_st:search_pt:premium_p:2_s:none_dt:desktop_reg:10658_ret:_apt:none&yclid=7649723942455005370 (Дата обращения 10.03.2022)
2. Мой ломбард [Электронный ресурс]: URL: <https://lombard-eps.ru/o-programme/avtomaticheskie-otchety-po-metrikam/> (Дата обращения 10.03.2022)
3. Что такое SQL Server: плюсы и минусы использования [Электронный ресурс]: URL: <https://muzeon.ru/medicina/2912-chto-takoe-sql-server-plyusy-i-minusy-ispolzovaniya.html> (Дата обращения 20.04.2022)
4. Моделирование данных: зачем нужно и как реализовать [Электронный ресурс]: URL: <https://habr.com/ru/post/554388/> (Дата обращения: 23.04.2022)
5. Концептуальная модель [Электронный ресурс]: URL: [http://wikiredia.ru/wiki/Концептуальная_модель#:~:text=Концептуальная%20модель%20\(англ.%20conceptual%20model\),модель%20—%20модель%20предметной%20области](http://wikiredia.ru/wiki/Концептуальная_модель#:~:text=Концептуальная%20модель%20(англ.%20conceptual%20model),модель%20—%20модель%20предметной%20области) (Дата обращения: 30.04.2022)
6. Логическая модель [Электронный ресурс]: URL: <https://teletype.in/@reddo/SJsmWPnpV#:~:text=Логическая%20модель%20—%20графическое%20представление,базы%20данных%20и%20аппаратной%20платформы> (Дата обращения: 30.04.2022)
7. Контекстная диаграмма [Электронный ресурс]: URL: <https://systems.education/context-diagram> (Дата обращения: 30.04.2022)

Приложение 1. Функциональная модель

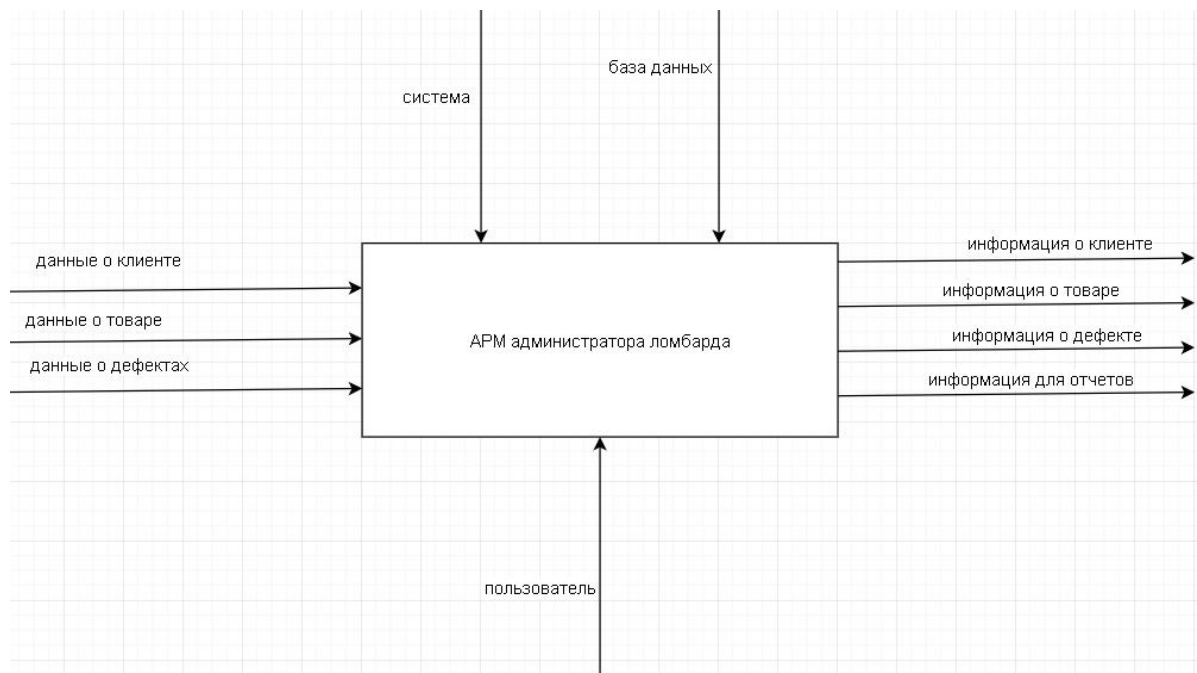


Рис. П1.1 - Главный процесс

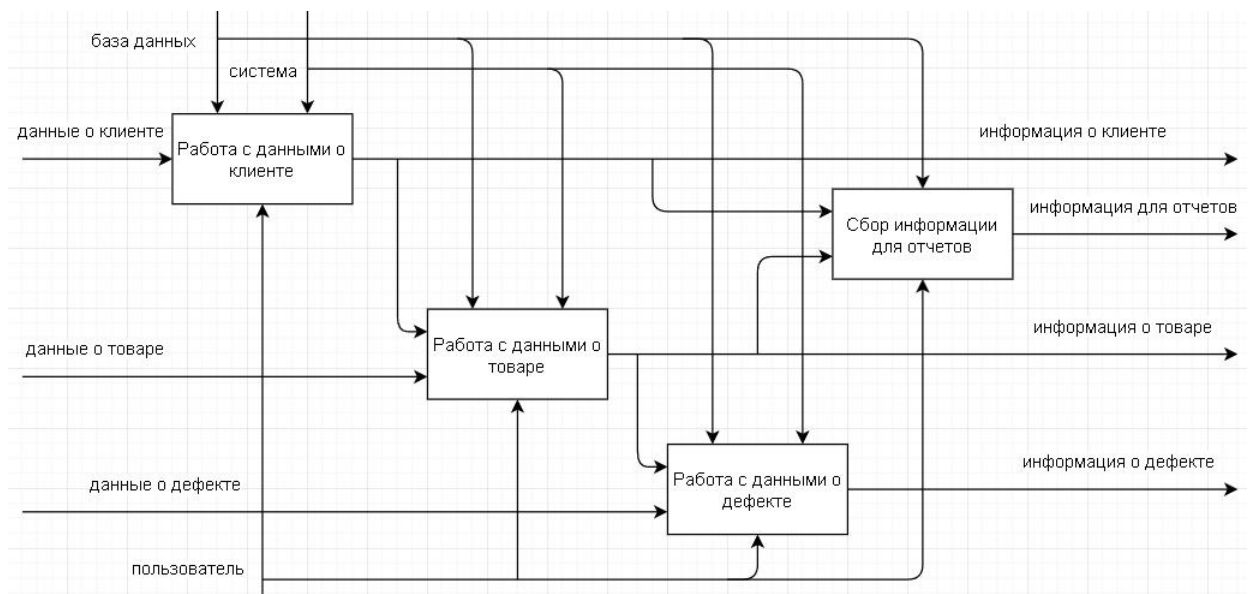


Рис. П1.2 - Декомпозиция главного процесса

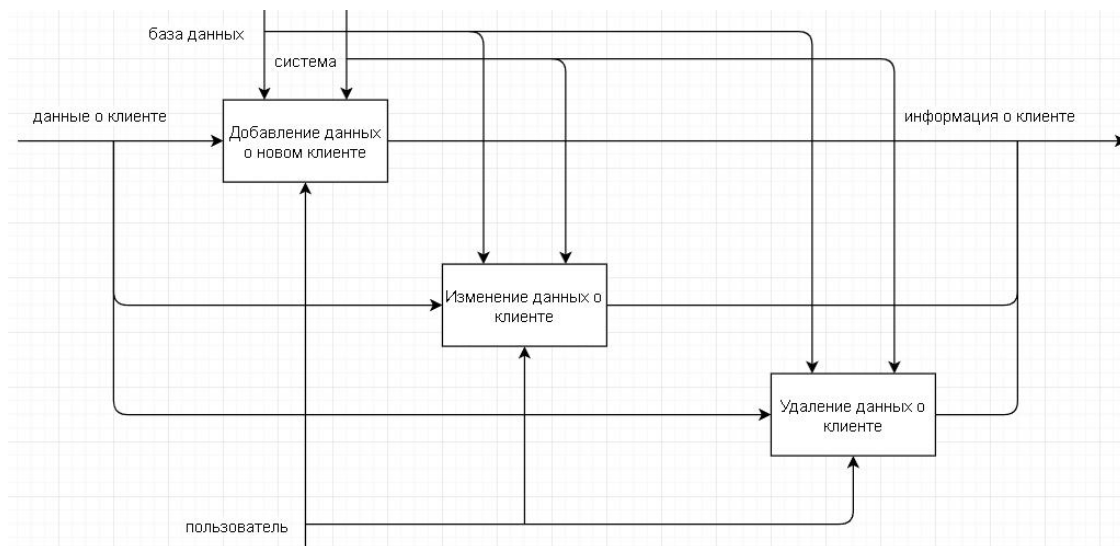


Рис. П1.3 - Декомпозиция процесса «Работа с данными о клиентах»

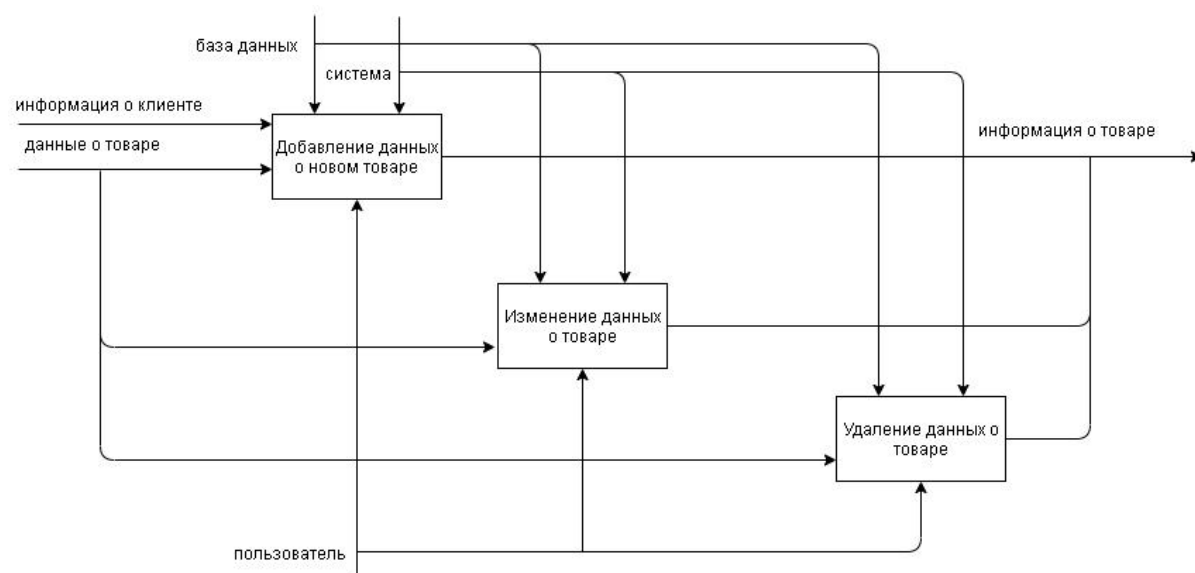


Рис. П1.4 - Декомпозиция процесса «Работа с данными о товарах»

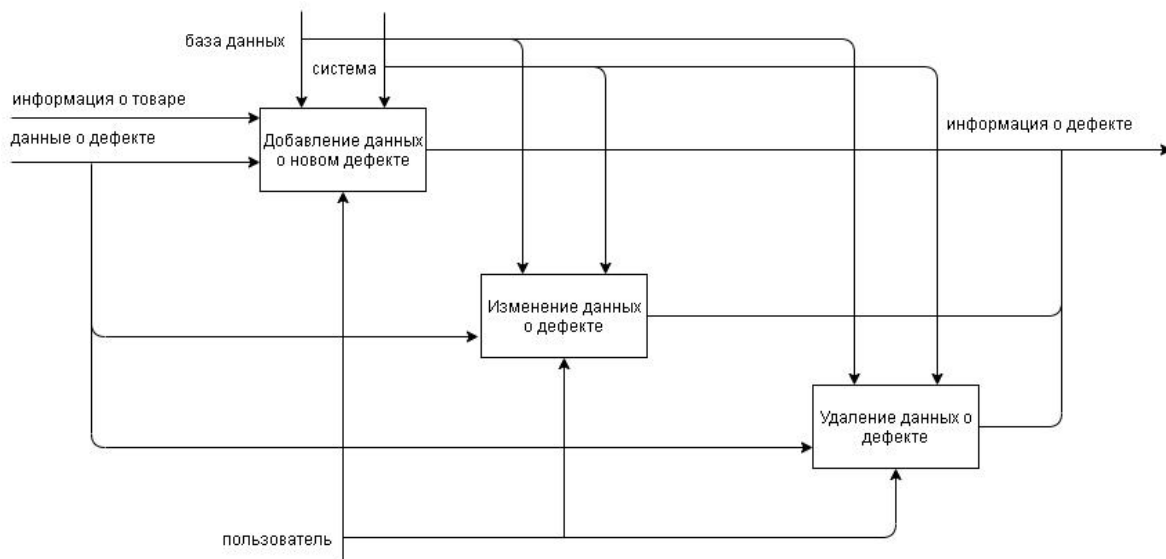


Рис. П1.5 - Декомпозиция процесса «Работа с данными о дефектах товара»

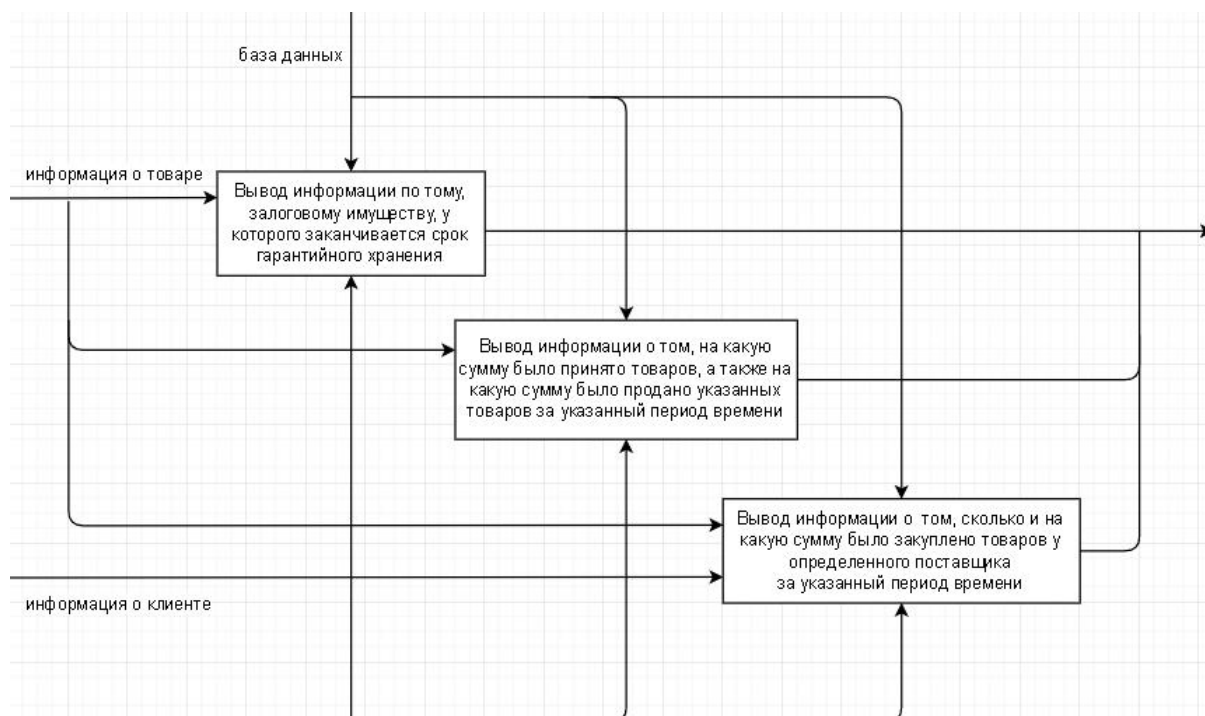


Рис. П1.6 - Декомпозиция процесса «Сбор информации для отчетов»

Приложение 2. Скриншоты программы

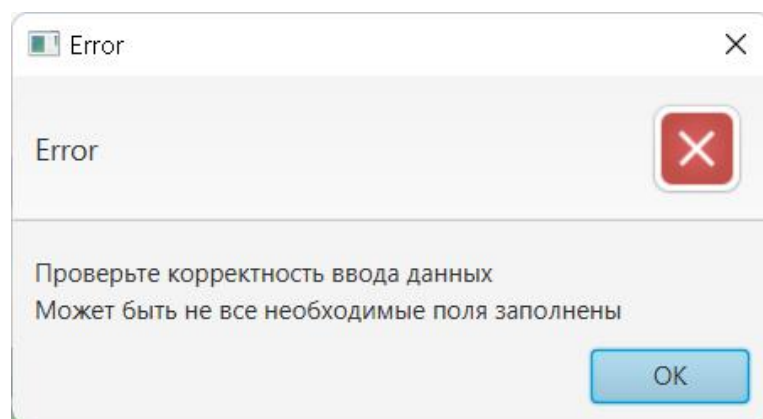


Рис. П2.1 – ошибка ввода данных клиента

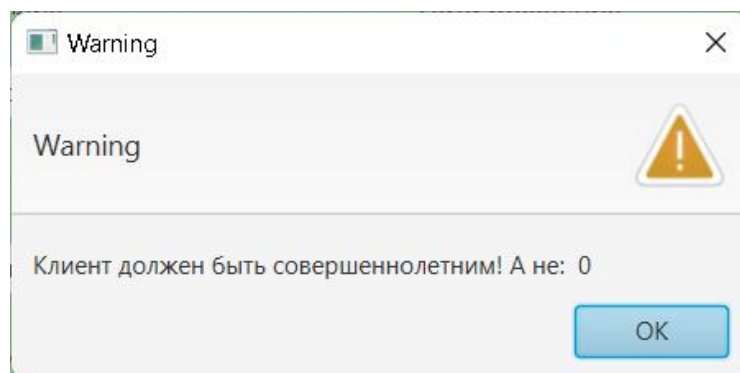


Рис. П2.2 – проверка на совершеннолетие по дате рождения

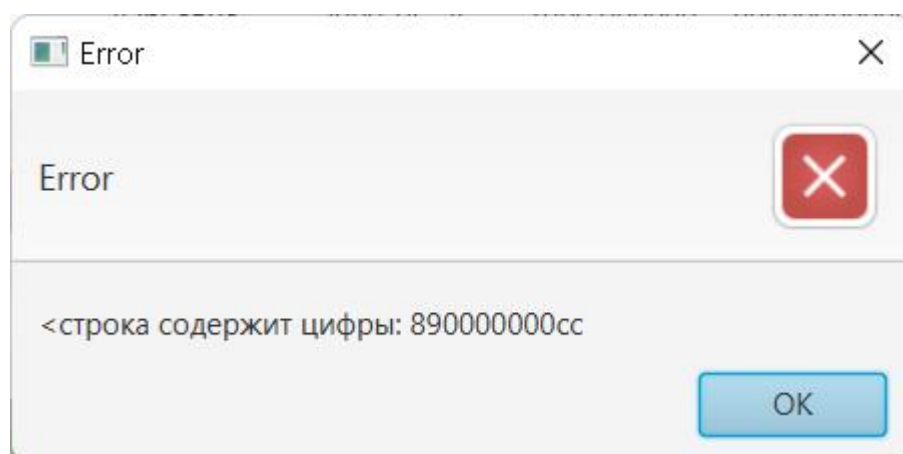


Рис. П2.3 – ошибка ввода номера телефона

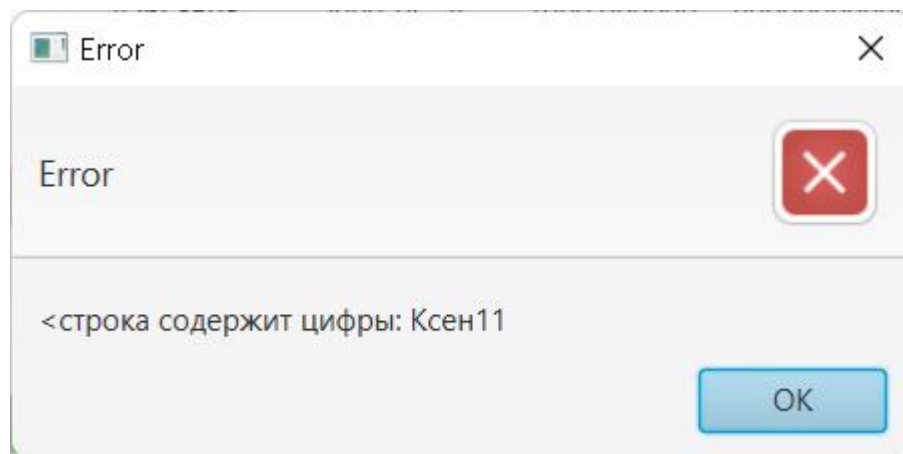


Рис. П2.4 – ошибка ввода ФИО

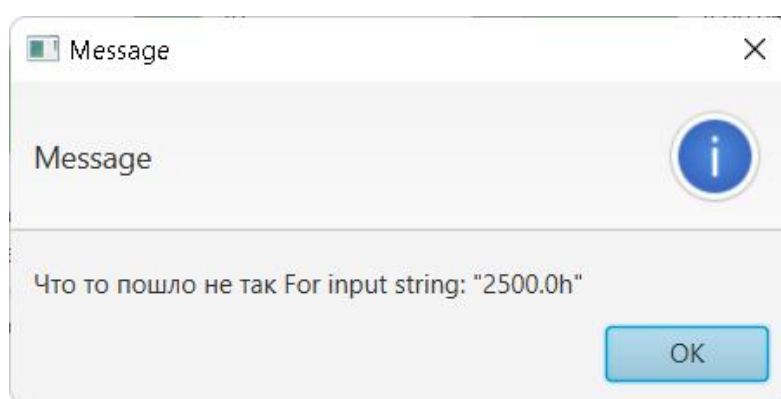


Рис. П2.5 – ошибка ввода цены товара

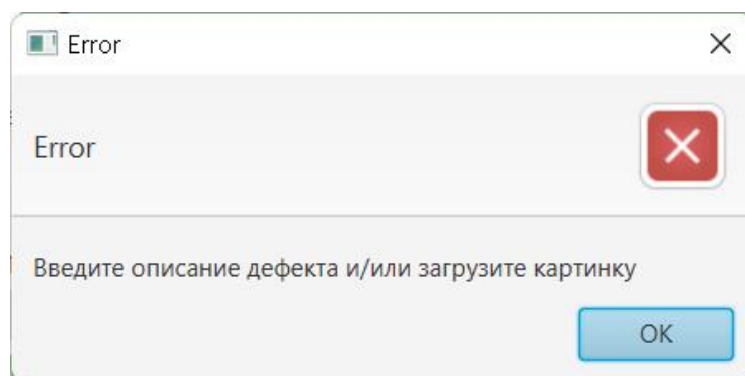


Рис. П2.6 – ошибка добавления дефекта

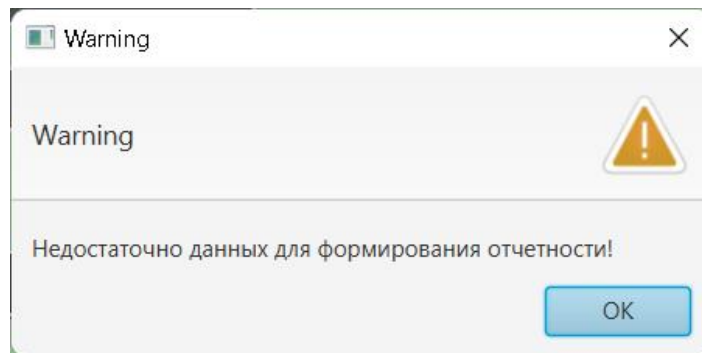


Рис. П2.7 – ошибка составления отчета

Приложение 3 Код программы

Код программы предоставлен в репозитории по ссылке:

<https://github.com/LvovaKs/Pavshop.iml.git>

					МИВУ.09.03.04-12.000 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подп.	Дата		