Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

«Колледж информационных технологий»

**ОТЧЕТ**

по учебной практике

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Специальность 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Выполнил |  |
| Студент гр. 493 | \_\_\_\_\_\_\_\_\_\_\_Э.Н. Баланин |
|  |  |
| Оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |
| Преподаватели УП | \_\_\_\_\_\_\_\_\_\_\_\_ И.А. Матысик |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Полякова  \_\_\_\_\_\_\_\_\_\_\_\_ И.П. Смирнова  \_\_\_\_\_\_\_\_\_\_\_\_ А.В. Фомин |

Санкт-Петербург 2023

**СОДЕРЖАНИЕ**

[1. УП 01.01. РАЗРАБОИКА ПРОГРАММНЫХ МОДУЛЕЙ 3](#_Toc129377777)

[1.1. Техническое задание 3](#_Toc129377778)

[1.2. Введение 4](#_Toc129377779)

[1.3. Общая часть 7](#_Toc129377780)

[1.3.1. Цель разработки 7](#_Toc129377781)

[1.3.2. Описание предметной области 7](#_Toc129377782)

[1.3.3. Анализ средств и технологий 8](#_Toc129377783)

[1.4. Проектная часть 11](#_Toc129377784)

[1.4.1. Постановка задачи 11](#_Toc129377785)

[1.4.2. Функционально-логическая структура проекта 12](#_Toc129377786)

[Диаграмма вариантов использования 12](#_Toc129377787)

[Диаграмма деятельности 13](#_Toc129377788)

[1.4.3. Логическая модель данных 14](#_Toc129377789)

[1.4.4. Разработка программных модулей 18](#_Toc129377790)

[1.4.5. Руководство пользователя 19](#_Toc129377791)

[2. УП01.02. Поддержка и тестирование программных модулей 29](#_Toc129377792)

[2.1. Практическая работа №1. Модульное тестирование. Составление Test-Case 29](#_Toc129377793)

[3. УП 01.03 РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ 43](#_Toc129377794)

[3.1. Анализ предметной области 43](#_Toc129377798)

[3.2. Создание макетов приложения 43](#_Toc129377799)

[3.3. Описание функций API 46](#_Toc129377800)

[3.4. Интерфейс пользователя 54](#_Toc129377801)

[3.5. Руководство пользователя 58](#_Toc129377802)

# **УП 01.01. РАЗРАБОИКА ПРОГРАММНЫХ МОДУЛЕЙ**

* 1. **Техническое задание**

Требуется разработать информационную систему (ИС) «Компьютерный Мир» – ПО разрабатываемое как для клиентов (с целью демонстрирования ассортимента компании, помощи в выборе товара и оформлении заказа), так и для автоматизации работы компании (добавление новых товаров, проведение заказов). Для этого требуется разработать основные программные модули ИС для следующих пользователей:

* + неавторизованный клиент и авторизованный клиент может просматривать товары и формировать заказ;
  + менеджер может просматривать товары, формировать и редактировать заказы;
  + системный администратор может добавлять/редактировать/удалять товары, просматривать и редактировать заказы.

Выполнить анализ и проектирование бизнес-процессов разрабатываемой информационной системы путем составления диаграммы прецедентов (Use–Case) для пользователей системы.

Выполнить проектирование реляционного хранилища данных путем составления ER-диаграммы разрабатываемой информационной системы и ведение словаря данных.

Выполнить разработку базы данных информационной системы «Компьютерный мир» на сервере баз данных, создать таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. Выполнить импорт подготовленных данных в разработанную базу данных информационной системы.

Выполнить разработку desktop-приложения для автоматизации работы сотрудников компании предоставляющих продажу компьютерных комплектующих. Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему, т.е. приложение должно обеспечивать процесс авторизации пользователя в системе.

Клиент может просматривать информацию о существующих товарах. Так же клиенту доступна возможность формирования заказа.

Менеджер обеспечивает создание заказа и также обладает возможностью просмотра номенклатуры компании.

Системный администратор обеспечивает безопасность системы от внешнего доступа к данным о сотрудниках и клиентах компании. В его функции входит контроль за входом в систему (ведение истории входов), выдача уникальных учетных данных для авторизации в системе.

В целом разрабатываемая система направлена на автоматизацию бизнес-процессов, протекающих в компании занимающейся продажей компьютерных комплектующих.

* 1. **Введение**

Учебная практика по МДК 01.01 «Разработка программных модулей» является обязательным учебным элементом для формирования общих и профессиональных компетенцией по модулю ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем».

Практика проходит в колледже информационных технологий, в компьютерной лаборатории №213 «Прикладное программирование». Период прохождения практики с 23.02.2023 по 15.03.2023. Объем учебной практики 30 часов. Тема учебной практики: «Разработка информационной системы магазина по продаже компьютерных комплектующих».

В отчете к учебной практике представлена краткая необходимая теоретическая справка, представлена четкая постановка задачи, представлены диаграммы, описывающие функциональную структуру программы и логическую модель данных, подробная инструкция пользователя и выводы по проделанной работе. Представлен программный код в приложении. В результате прохождения учебной практики по МДК 01.01 были подтверждены общие и профессиональные компетенции.

Перечень общих компетенций:

* + ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам;
  + ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;
  + ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие;
  + ОК 4. Планировать и реализовывать собственное профессиональное и личностное развитие;
  + ОК 5. Планировать и реализовывать собственное профессиональное и личностное развитие;
  + ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения;
  + ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях;
  + ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;
  + ОК 9. Использовать информационные технологии в профессиональной деятельности;
  + ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

Перечень профессиональных компетенций:

* + ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;
  + ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием;

В результате прохождения учебной практики по МДК 01.01 «Разработка программных модулей», был получен практический опыт в следующих пунктах:

* + в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
  + в использовании инструментальных средств на этапе отладки программного продукта.

В результате прохождения учебной практики, были получены умения:

* + осуществлять разработку кода программного модуля на языке высокого уровня;
  + создавать программу по разработанному алгоритму как отдельный модуль;
  + осуществлять разработку кода программного модуля на современных языках программирования;
  + выполнять оптимизацию и рефакторинг программного кода;
  + оформлять документацию на программные средства;
  + разрабатывать код программного модуля с применением технологии объектно-ориентированного программирования;
  + использовать современные технологии обработки данных.
  + в результате прохождения учебной практики по модулю, были получены знания по следующим понятиям:
  + основные этапы разработки программного обеспечения;
  + основные принципы технологии структурного и объектно-ориентированного программирования;
  + основные возможности современных технологий.
  1. **Общая часть**

В данной части отчета по учебной практике представлена краткая теория по теме учебной практике, а также представлены результаты исследования проведенного анализа программных средств и технологий для разработки проекта учебной практики.

* + 1. **Цель разработки**

Целью проекта является проектирование и разработка информационной системы для клиентов и специалистов компании по продаже компьютерных комплектующих, которая поможет автоматизировать добавления новых товаров, п, что сократит время обслуживания клиента компании. В итоге разрабатываемая система должна привлечь больше потенциальных клиентов, обеспечив стабильный рост доходов и рейтингов компании, для которого разрабатывается данная информационная система.

* + 1. **Описание предметной области**

Информационная система "Компьютерный мир" – прикладное программное обеспечение, использующееся для предоставления клиентам сведений об товаре выставляемом на продажу компанией, а также ряда следующих функциональных особенностей:

* + добавление нового товара в базу данных и предоставление сведений о нем клиентам;
  + редактирование или удаление уже существующего товара, оформление заказа для клиента.

Основными целями внедрения системы являются:

* + упрощение предоставления информации об ассортименте компании;
  + автоматизация процесса оформления заказа;
  + минимизация человеческого фактора при ведении документов;
  + снижение рутиной работы при работе с документами.

Рассмотрим определение прецедентов (вариантов использования). Система требуется, прежде всего, следующим заинтересованным лицам:

* + менеджер компании;
  + клиент;
  + системный администратор.

Функционал каждой роли:

Системный администратор и менеджер обладают возможностью просмотра, редактирования, добавления, удаления данных. Ко всему прочему системный администратор имеет возможность создавать пользователей для системы, удалять их или же изменять их учётные данные.

Клиент может только просматривать сведения о товаре и формировать заказы.

Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему.

Проект предназначен для коммерческого компании, занимающегося продажей компьютерных комплектующих физическим и юридическим лицам.

Внедрение данного проекта поможет автоматизировать процесс добавления новых клиентов, способствуя снижению рутиной работы с ведением документооборота и минимизации случаев человеческого фактора.

* + 1. **Анализ средств и технологий**

В данном разделе описываются возможные пути и программные средства, с помощью которых можно реализовать приложение.

В данном подразделе пояснительной записки представлен материал проведенного анализа программных средств, которыми можно разработать приложение. В анализе присутствуют сравнительные характеристики программных технологий и методов, языков программирования и интегрированных сред.

Инструментальные средства:

* + операционная система – Windows 10;
  + язык программирования – "C#";
  + среда программирования – "Microsoft Visual Studio 2022";
  + СУБД – "SQL Server Management Studio 2019".

Для создания приложения будет использована интегрированная среда программирования "Microsoft Visual Studio 2022", язык программирования высокого уровня С#, систему построения клиентских приложений "Windows Presentation Foundation" (WPF), а также "SQL Server Management Studio 2019" для создания базы данных.

В качестве среды разработки была выбрана инструментальная среда разработки "Microsoft Visual Studio 2022", включающая в себя интегрированную среду разработки, редактор исходного кода, встроенный отладчик. Многие другие инструменты возможно получить благодаря подключению плагинов – сторонних расширений. Был выбран по причине поддержки языка программирования С#. Также это наиболее изученная разработчиком платформа, что позволяет быстрее разрабатывать приложения не тратя время на изучения нового материала.

В качестве языка разработки был выбран язык программирования C# – объектно-ориентированный язык программирования. Разработан в 1998 — 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы "Microsoft .NET Framework".

Достоинства С# по книге Биллига:

* + создавался параллельно с каркасом ".Net Framework" и в полной мере учитывает все его возможности - как FCL, так и CLR;
  + является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
  + является мощным объектным языком с возможностями наследования и универсализации;
  + является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход от С++ к C#;
  + сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика;
  + благодаря каркасу ".Net", ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java;
  + мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
  + реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

Выделение и объединение лучших идей современных языков программирования делает язык C# не просто суммой их достоинств, а языком программирования нового поколения.

"Windows Presentation Foundation" (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе ".NET Framework" (начиная с версии 3.0), использующая язык XAML.

"SQL Server Management Studio" – утилита из "Microsoft SQL Server 2019" и более поздних версий для конфигурирования, управления и администрирования всех компонентов "Microsoft SQL Server". Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

* 1. **Проектная часть**

В данной части отчета по учебной практике представлено подробное описание разработанного приложения, соответствующего поставленной задаче в техническом задании к учебной практике.

* + 1. **Постановка задачи**

Информационная система «Компьютерный мир» – прикладное программное обеспечение, использующееся для предоставления информации о товарах предоставляемых на продажу компанией, упрощения формирования заказа пользователями, а также предназначена для решения следующих задач:

* + контроль заказов;
  + формирование необходимой информации по итогам оформленных заказов на физическое или юридическое лицо (дата оформления заказа, количество товаров в заказе, необходимая для выплаты сумма).

Основными целями внедрения системы являются:

* + упрощение предоставления информации об ассортименте компании;
  + автоматизация процесса оформления заказа;
  + минимизация человеческого фактора при ведении документов.

Исходная информация в систему поступает из документов о номенклатуре компании. Входными данными являются следующие:

* + название товара;
  + изображение товара;
  + системные характеристики товара.

Рассмотрим определение прецедентов (вариантов использования). Система требуется прежде всего следующим заинтересованным лицам:

* + менеджер;
  + клиент;
  + системный администратор.

Определяется перечень функциональных подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы.

Смежными системами для «Клиенты» являются:

* + информационные системы ввода, хранения, поиска и оперативной обработки информации о клиентах;
  + информационная система формирование отчетов, необходимых менеджеру, содержащих следующие данные о клиенте: ФИО, номер телефона.

Источниками данных для системы должны быть:

* + СУБД MS SQL;
  + таблицы Excel.

Входными данными являются следующий набор данных:

* + ФИО клиента;
  + номер телефона клиента;
  + логин и пароль.

Система должна функционировать в многопользовательском режиме и давать возможность:

* + Просматривать записи базы данных, в том числе и с помощью различных фильтров;
  + Добавлять новые записи и изменять существующие;
  + Удалять записи.
    1. **Функционально-логическая структура проекта**

В данном разделе отчета по учебной практике представлены различные диаграммы на языке UML, описывающие функциональную и логическую структуру разрабатываемой программы учебной практики.

**Диаграмма вариантов использования**

В этом разделе приведена UML-диаграмма типа вариантов использования или диаграмма прецедентов, представленная на рисунке 1. Диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Прецедент - возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

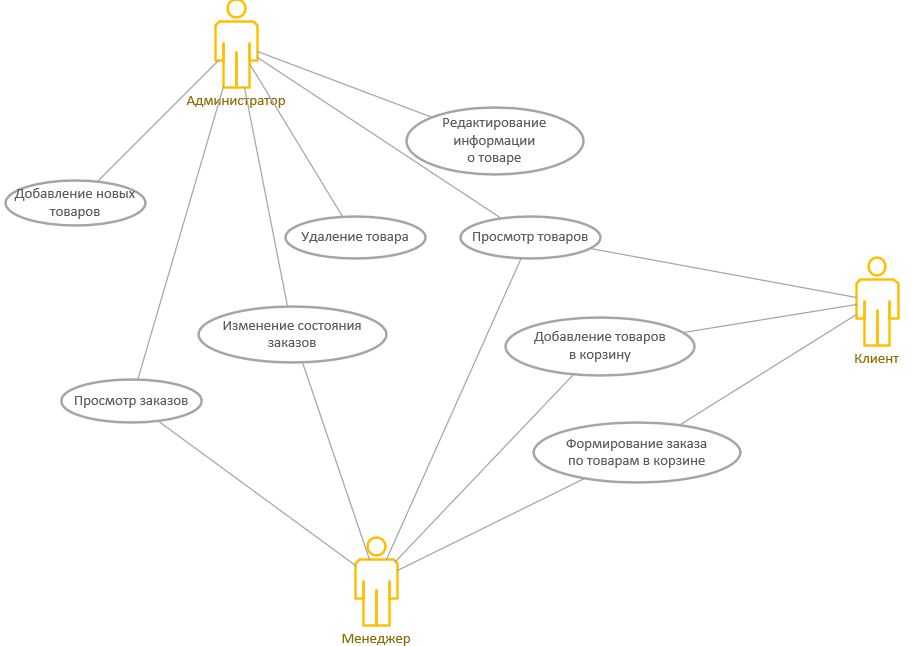
****

Рисунок 1 – Диаграмма вариантов использования

**Диаграмма деятельности**

Диаграмма деятельности (англ. "activity diagram") – UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Была разработана диаграмма деятельности для прецедента «Добавление нового товара», которая осуществляется ролью «системный администратор». Данная диаграмма представлена на рисунке 2.

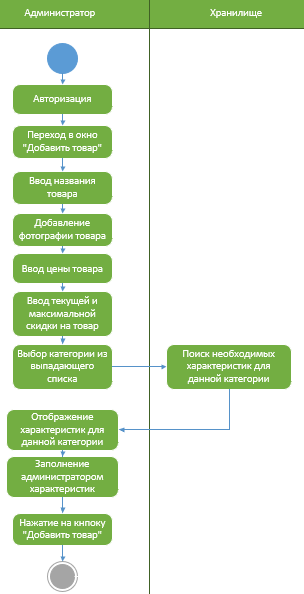


Рисунок 2 – Диаграмма деятельности «Добавление нового товара»

* + 1. **Логическая модель данных**

В основе разрабатываемого приложения находится база данных, как структурированное хранилище информации разрабатываемой системы. Удобно логическую модель данных описывать с применением UML-языка на основе ED-диаграммой. На рисунке 3 представлена ER-диаграмма разработанной базы данных.

В системе есть две сущности: «Пользователи» и «Товары» и «Характеристики».

Таблица "User" состоит из полей Фамилия, Имя, Отчество, email-адрес, логин, пароль и ID роли (связанная таблица, описана ниже).

Таблица "Role" состоит из полей идентификатора записи и названия роли.

Таблица "Product" состоит из полей идентификатора, названия товара, идентификатора категории данного товара (внешний ключ, связывающий данную таблицу с таблицей «Category», описана ниже), цены товара, текущей и максимальной скидки на товар.

Таблица "Category" состоит из следующих полей: поле-идентификатор записи, название категории в единственном числе.

Таблица "ProductPicture" содержит в себе поле-идентификатор записи, идентификатор продукта (при помощи этого поля таблица «ProductPicture» связана с таблицей «Product») и поле двоичного представления изображения данного товара.

Таблица "Characteristic" содержит в себе поле идентификатор записи и название характеристики (у разных товаров разные характеристики, при помощи данной таблицы получается гибко указывать к товарам характеристики).

Таблица "CategoryCharacteristic" обладает полем-идентификатором категории и полем идентификатором характеристики товара. Необходимость в данной таблице возникает во время добавления нового товара, нужно вывести определенный список характеристик для их дальнейшего заполнения администратором.

Таблица "ProductCharacteristic" содержит в себе поле-идентификатор записи, поле-идентификатор товара к которому относится данная запись, поле-идентификатор характеристики, значение этой самой характеристики и поле для записи единиц измерения.

Таблица "Outpost" состоит из поля-идентификатора и поля содержащего в себе адрес пункта выдачи товара.

Таблица "OrderStatus" содержит в себе поле-идентификатор записи и поле содержащее в себе возможное состояние заказа (новый, завершен и другие возможные состояния).

Таблица "Order" содержит в себе информацию о заказе: дата создания заказа, приблизительная дата получения заказа, поле содержащее идентификатор пункта выдачи (связь с таблицей "Outpost"), поле содержащее в себе идентификатор состояния заказа (связь с таблицей "OrderStatus"), код для получения заказа в пункте выдачи, ФИО клиента оформившего заказ (поле может быть пустым в случае если заказ сформировал гость).

Таблица "OrderContent" состоит из поля-идентификатора продукта, поля-идентификатора заказа и поля отвечающего за кол-во заказанного товара конкретной позиции.

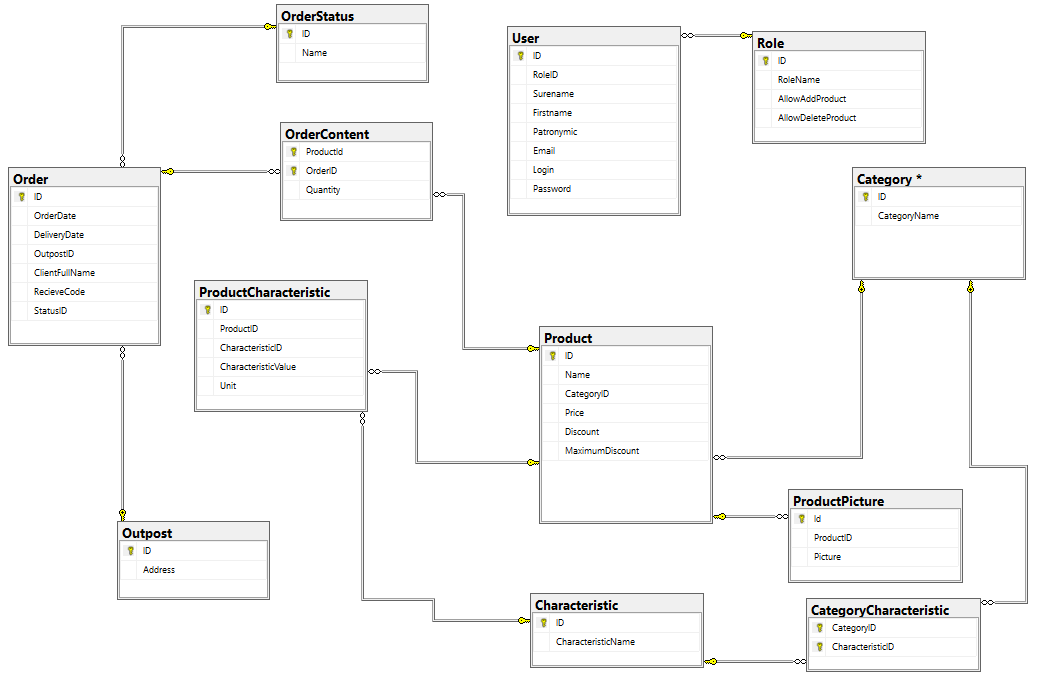


Рисунок 3 – ER-диаграмма базы данных

На рисунке 4 представлен заполненный словарь данных системы – набор информации, описывающий, какой тип данных хранится в БД, их формат, структуру и способы использования данных. Дополнительно в словаре данных могут содержаться описания к отдельным полям таблицы базы данных.



Рисунок 4 – Словарь данных системы

На рисунке 5 изображен словарь данных для таблиц «ProductCharacteristic», «Outpost», «OrderStatus», «Order», «OrderContent».

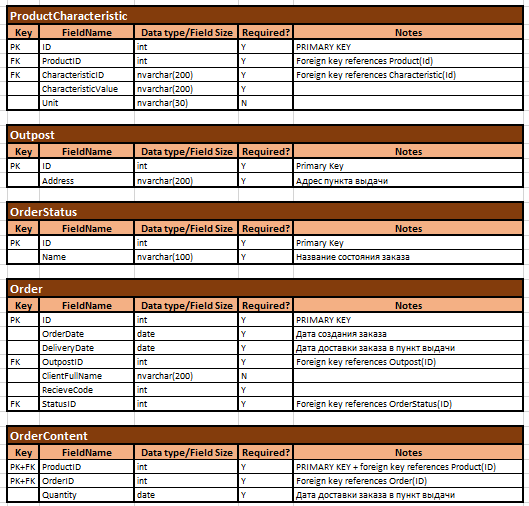


Рисунок 5 – Словарь данных для оставшихся таблиц

* + 1. **Разработка программных модулей**

В данном разделе отчета представлены действия, необходимые для программной реализации разрабатываемого приложения, а также требования к разработке пользовательского интерфейса.

**Пользовательский интерфейс**

Не допускаются орфографические и грамматические ошибки.

Все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом logo.png (расположен в ресурсах). Нельзя искажать логотип (не изменяйте изображение, его пропорции, цвет).

Также для приложений должна быть установлена иконка.

В случае возникновения ошибок система оповещает пользователя всплывающими окнами с информацией об ошибке.

* + 1. **Руководство пользователя**

Для начала работы необходимо запустить exe файл – PcComponents.exe

**Форма ввода данных для входа**

После запуска программы открывается окно аутентификации (рисунок 6), где нужно ввести логин и пароль для входа в систему, на рисунке 6 продемонстрировано данное окно.

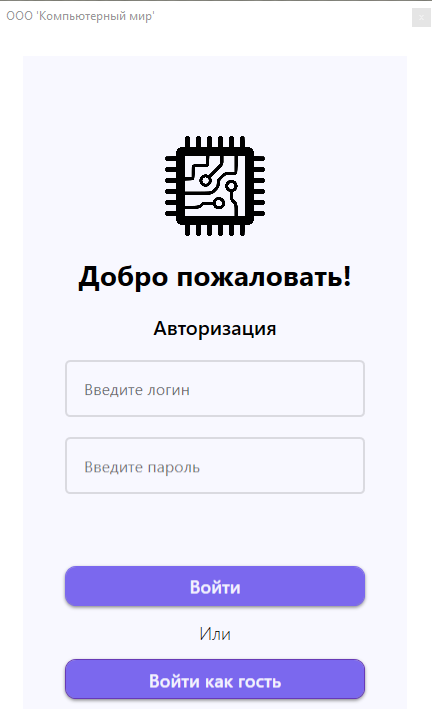


Рисунок 6 – Форма для ввода аутентификационных данных

При ошибочном вводе пароля или логина вам будет показано сообщение о том, что вы должны проверить правильность ввода, как показано на рисунке 7. Сообщение было показано, отслеживание ошибок ввода пароля работает.

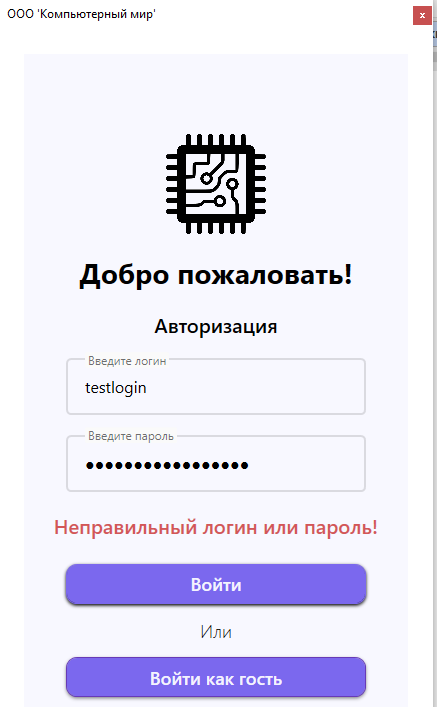


Рисунок 7 – Результат неправильного ввода пароля

В случае корректного ввода данных, программа сразу же откроет окно каталога для пользователя, под которым вы зашли. В данном случае это администратор, как показано на рисунке 8.

На рисунке 8 продемонстрирован результат успешной авторизации под учетную запись администратора

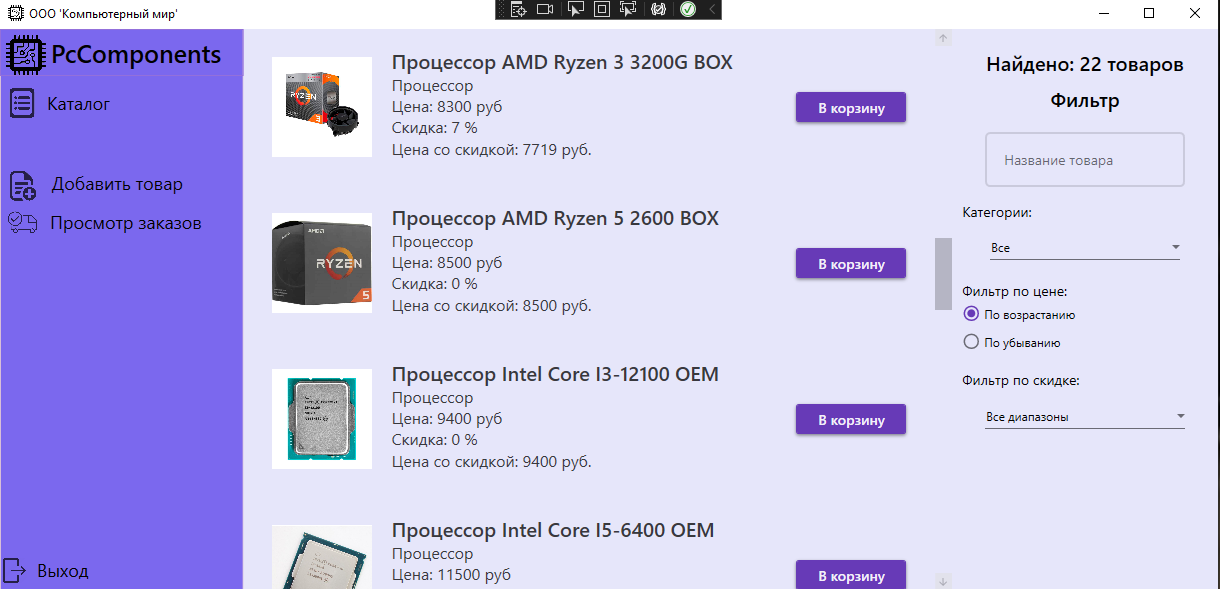


Рисунок 8 – Успешная авторизация, переход на окно каталога

**Каталог товаров**

Форма каталога товаров доступна каждому пользователю, но все роли кроме администратора могут формировать заказа по нажатию кнопки «В корзину». У администратора кнопка не функционирует.

В левой части экрана расположено навигационное меню при помощи которого можно перемещается по приложению. Здесь будут доступные следующее возможности: открытие каталога, открытие корзины, открытие окна добавление нового товара, просмотр существующих а также выход обратно к окну авторизации. Все эти переходы разграничены для каждой роли пользователя в соответствии обозначенными вариантами использования в диаграмме вариантов использования (рисунок 1) (кнопка для перехода в корзину отсутствует, т.к. авторизация была выполнена на роль системного администратора, у которого нет возможности формирования заказов).

Каталог товаров можно отсортировать по цене товара: по возрастанию или убыванию. Также доступна сортировка товара по диапазонам скидки: все диапазоны, от 0 до 3%, от 3 до 6%, 7% и более. Также доступен поиск товара по его имени и категории. В верхней правой части экрана отображается итоговое кол-во найденных товаров по всех описанным выше категориям. Если в поле для поиска товара по названию введена пустая строка или ничего не введено, то поиск по имени товара не осуществляется. По нажатию на кнопку «В корзину» товар добавится в корзину (только у гостя, авторизированного клиента и менеджера). В левой части экрана около слова «Корзина» появится число означающее кол-во разных товаров помещенных в корзину.

На рисунке 9 продемонстрирован поиск товаров по категории «Процессор», содержащие в названии «Intel» отсортированные по убыванию цены.

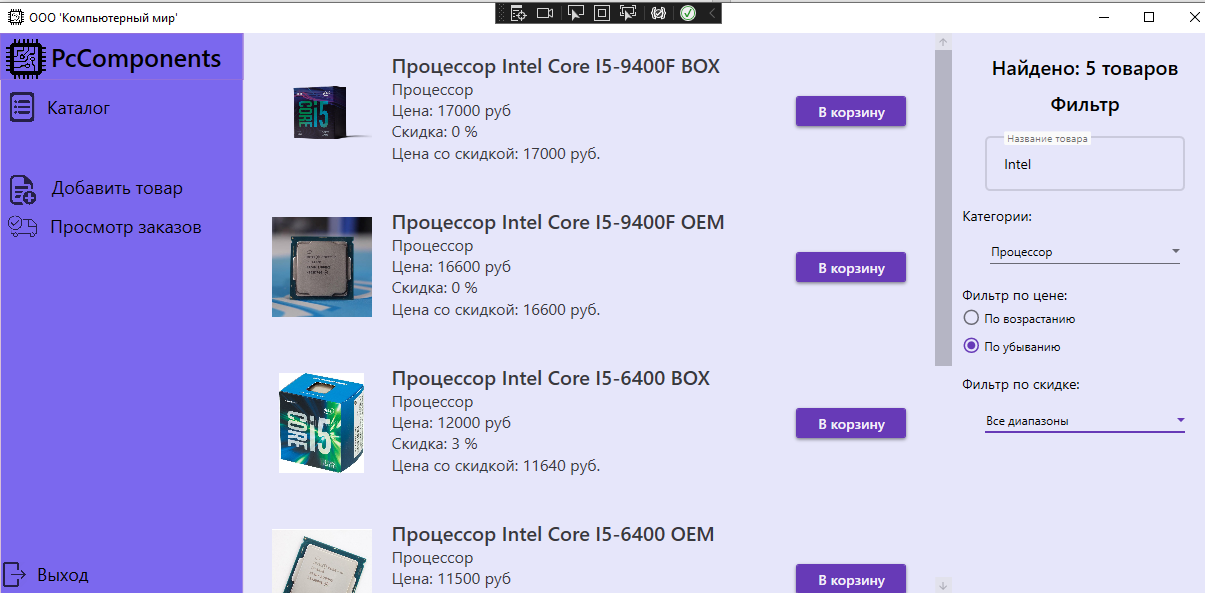


Рисунок 9 – Отфильтрованный каталог

**Окно корзины**

Для добавления товара в корзину необходимо выбрать в списке товаров интересующий вас товар и нажать на кнопку «В корзину». Далее в левом меню около слова «Каталог» появится цифра символизирующая кол-во различных товаров добавленных в вашу корзину. По нажатию на эту кнопку «Каталог» вы перейдете в окно каталога, где будут отображены все выбранные вами товары, ваше ФИО или слово гость в случае авторизации в качестве гостя, общая цена за все товары с учетом кол-ва каждого вида товара.

Если вам необходимо несколько товаров одного типа, то увеличить кол-во выбранного в корзине товара можно будет при помощи кнопок «->» или уменьшить при помощи кнопки «<-». Также между двух ранее указанных кнопок находится поле для ввода кол-ва товара. В случае если в этом поле число товара становится 0, то товар исчезает из корзины. Общая цена обновляется при любом изменении корзины, добавление нового товара, удалении товара или изменении кол-ва товаров одного вида. На рисунке 10 изображено меню корзины пользователя роли «Клиент».

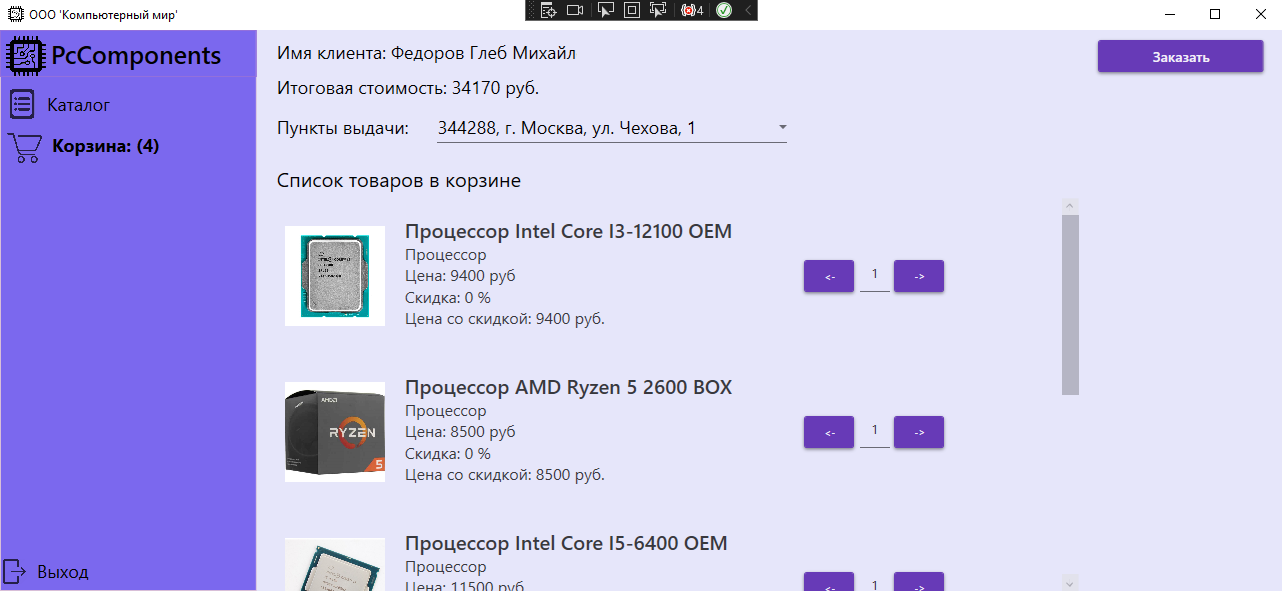


Рисунок 10 – Окно корзины

При нажатии на кнопку «Заказать» в верхнем правом углу приложения, сформируется заказ в выбранный пункт выдачи и на экране появится сообщение показанное на рисунке 11.

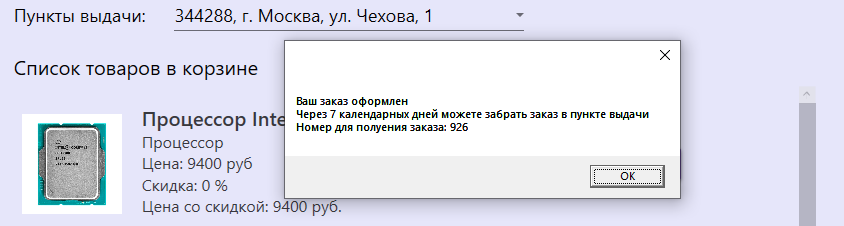


Рисунок 11 – Оформление заказа

Номер для получения заказа уникальный для каждого заказа, его необходимо запомнить, при помощи этого номер придя в пункт выдачи вы заберете свой заказ (также можно идентифицировать заказ по своему ФИО в случае оформления заказа под ролью «Клиент»).

**Просмотр полной информации о товаре**

Находясь в каталоге можно посмотреть подробную информацию о товаре дважды нажав на него левой кнопкой мыши. На рисунке 12 изображено окно просмотра полной информации о товаре.



Рисунок 12 – Полная информация о товаре

**Просмотр заказов**

Для пользователей с ролями «Менеджер» и «Администратор» доступен просмотр оформленных заказов. В данном окне доступна для просмотра большая часть информации о заказе: Номер заказ, ФИО клиента оформившего заказ (или пустая строка в случае оформления заказа гостем), адрес пункта выдачи, дата формирования заказа и дата доставки заказа в пункт выдачи. Также доступна возможность для поиска заказа по его номеру в правом верхнем углу окна просмотра заказов. При нажатии на определенный заказ будет выполнен переход на форму содержащую подробную информацию о данном заказе (в том числе товары данного заказа).

На рисунке 13 продемонстрировано окно для просмотра заказов. Отображены только товары которые удовлетворяют значению в поле поиска товара по его номеру (в поле записано значение 1). Для завершенных заказов кнопка «Изменить» недоступна, т.к. подразумевается что заказ уже получен клиентом.

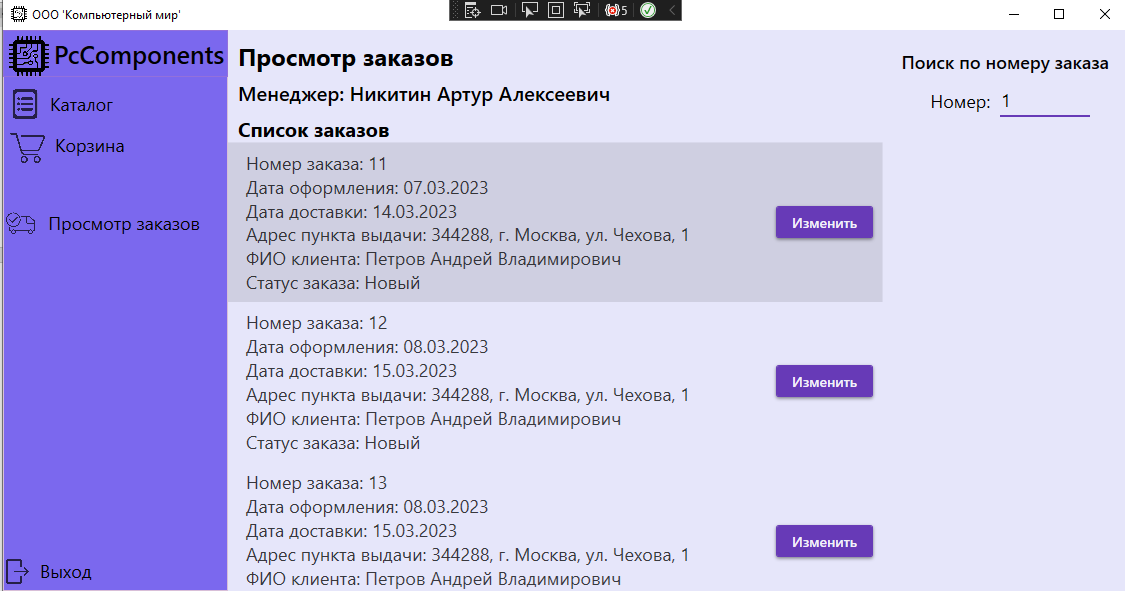


Рисунок 13 – Окно просмотра заказов

При двойном нажатии на конкретный заказ произойдет переход на окно с подробным его описанием. На рисунке 14 продемонстрировано данное окно. При двойном нажатии на товар будет открыто окно с его подробным описанием. Данное окно идентично окну, продемонстрированному на рисунке 12.

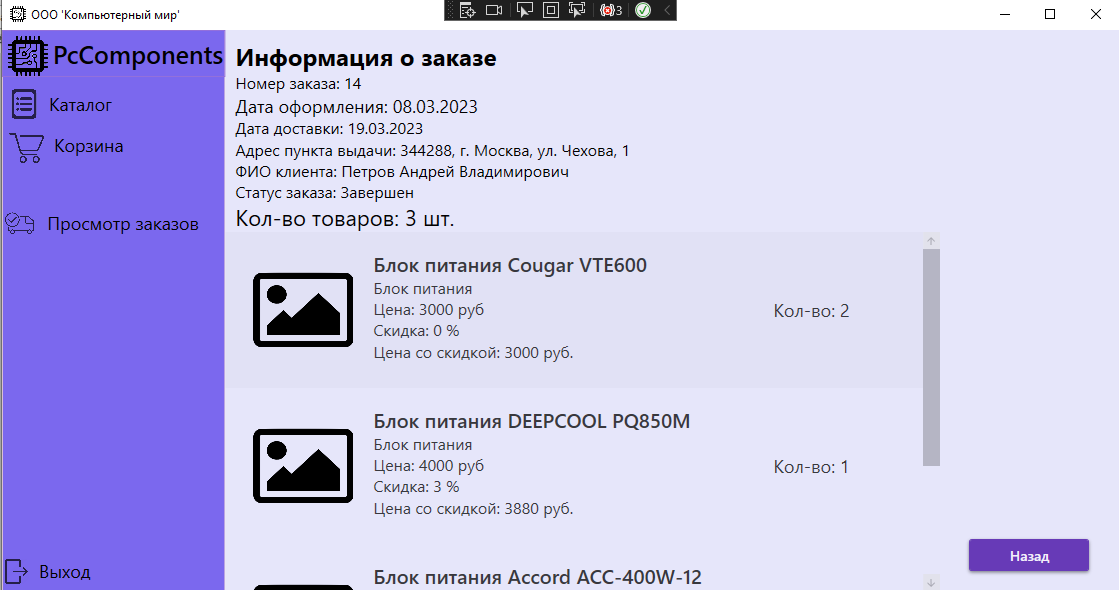


Рисунок 14 – Подробное описание заказа

При нажатии на кнопку «Изменить» открывается окно подробной информации о заказе, где становятся доступные для редактирования поля даты формирования заказ, дата доставки заказа, ФИО клиента и состояние заказа. Все изменения по нажатию на кнопку «Сохранить» применятся и отразятся в базе данных. На рисунке 15 изображено данное окно редактирования заказа.

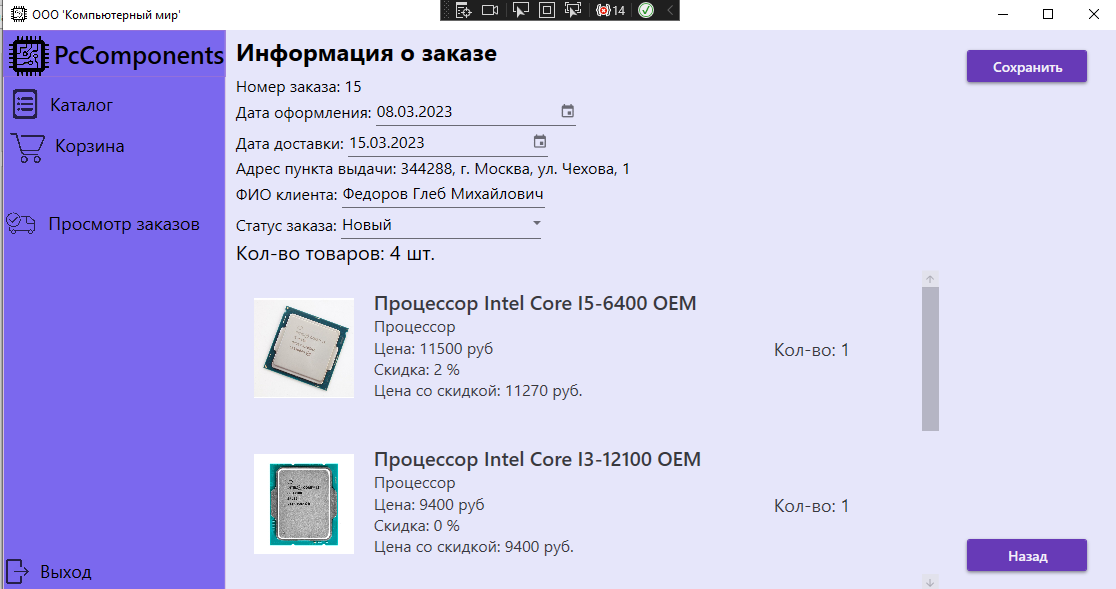


Рисунок 15 – Окно редактирования заказа

**Добавление нового товара и редактирование информации о товаре**

Администратору доступна функция добавления нового товара. По нажатию на кнопку «Добавить товар» в левом навигационном меню приложения произойдет переход на новое окно добавления нового товара. Становятся доступны поля «Название товара», выпадающий список для выбора категории нового товара, поле для указания цены нового товара, текущей скидки и поле для максимальной скидки товара. Также есть возможность добавления картинки к товару по нажатию на кнопку «Изменить картинку». По умолчанию картинки у нового товара нет, выводится заглушка для данного товара которая будет отображаться в случае если картинка не будет добавлена. Нажатие на кнопку «Изменить картинку» открывает диалоговое окно для выбора картинки с компьютера на котором находится приложение. При выборе картинки она автоматически отобразится на соответствующем компоненте отображения и запишется к новому пока еще не созданному товару.

При выборе категории товара автоматически загрузятся общие характеристики для данной категории товаров. Не обязательно заполнять все характеристики, в случае если некоторые из них были пропущены они не запишутся в базу данных и товар не будет владеть информацией о них.

На рисунке 16 изображено окно для создания нового товара категории «Процессоры» где уже заполнены некоторые характеристики данного товара.

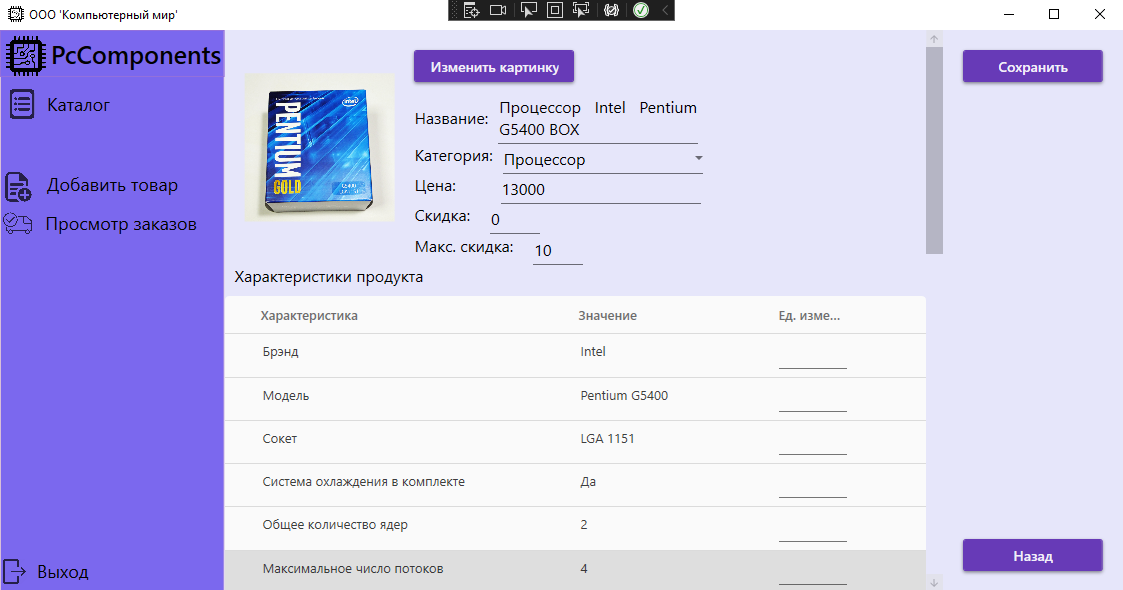


Рисунок 16 – Добавление нового товара

При нажатии на кнопку «Сохранить» отобразится сообщение «Товар успешно добавлен» и данные о новом товаре занесутся в БД и он станет доступен для отображения в каталоге.

Также для администратора доступна возможность редактирования информации о существующем товаре. Чтобы попасть в соответствующее окно администратору необходимо найти интересующий его товар в каталоге и дважды нажать на него. После этого, если авторизированный пользователь обладает ролью «Администратор», то откроется окно редактирования информации о данном товаре, где будут доступны для редактирования все поля.

На рисунке 17 продемонстрировано окно редактирования информации о товаре.

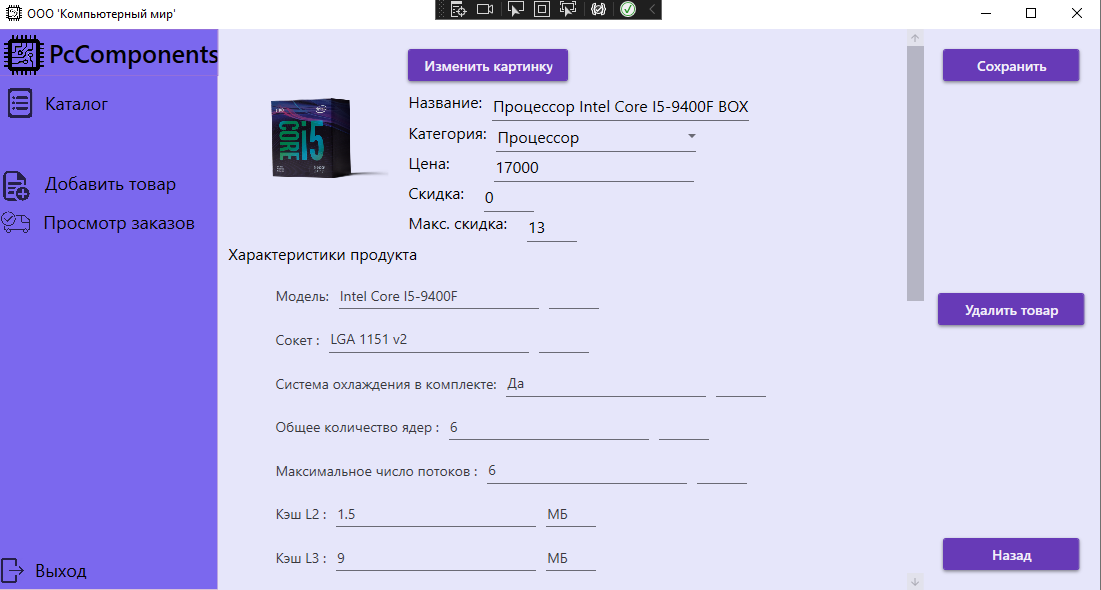


Рисунок 17 – Редактирование информации о товаре

При изменении какой-либо характеристики продукта или его основной информации или картинки и нажатии на кнопку «Сохранить» изменения автоматически отобразятся в базе данных и каталоге соответственно.

Также администратору доступна возможность удаления существующего товара. Для этого администратору необходимо найти интересующий его товар в каталоге, нажать на него дважды левой кнопкой мыши. После это откроется окно редактирования заказа, где доступна кнопка «Удалить товар». При нажатии на кнопку появится окно с подтверждением действия. В случае согласия подтвердить свои действия товар удалится из базы данных и каталога соответственно.

# **УП01.02. Поддержка и тестирование программных модулей**

## Практическая работа №1. Модульное тестирование. Составление Test-Case

Цель: провести модульное тестирование функциональности проекта и составить "Test-Case" к данной функциональности.

Задание: Класс расчёта материалов. Метод должен рассчитывать целое количество сырья, необходимого для производства определенного количества (count) продукции, учитывая возможный брак материалов. Для упрощения расчетов будем считать всю продукцию прямоугольного размера с известными значениями ширины (width) и длины (length).

Ссылка на репозиторий "github":

https://github.com/bl1st/Practice\_Testing\_Week1

Количество необходимого качественного сырья на одну единицу продукции рассчитывается как площадь продукции, умноженная на коэффициент типа продукции.

Коэффициенты типа продукции (product\_type):

* Тип продукции 1 - 1.1;
* Тип продукции 2 - 2.5;
* Тип продукции 3 - 8.43.

При этом нужно учитывать процент брака материала в зависимости от его типа (material\_type):

* Тип материала 1 - 0.3%;
* Тип материала 2 - 0.12%.

При этом если в качестве параметров метода будут приходить несуществующие типы продукции/материалов или другие неподходящие данные, то метод должен вернуть -1.

Например, необходимо изготовить 15 единиц продукции 3 типа шириной 20 и длиной 45 из материала 1 типа. Количество качественного сырья (без учета брака) будет равно 113 805. Однако с учетом возможного брака материалов общее необходимое количество сырья должно быть увеличено до 114 147,442. Округлив полученное значение до ближайшего большего целого, получим 114 148 единиц необходимого сырья.

Спецификация разрабатываемого метода предоставлена на рисунке 1.

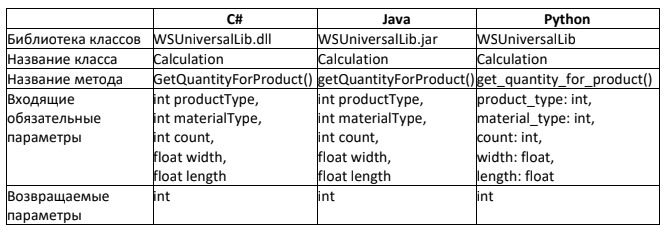


Рисунок 1 – Спецификация разрабатываемого метода

Программный код разработанной функциональности продемонстрирован на рисунке 2.

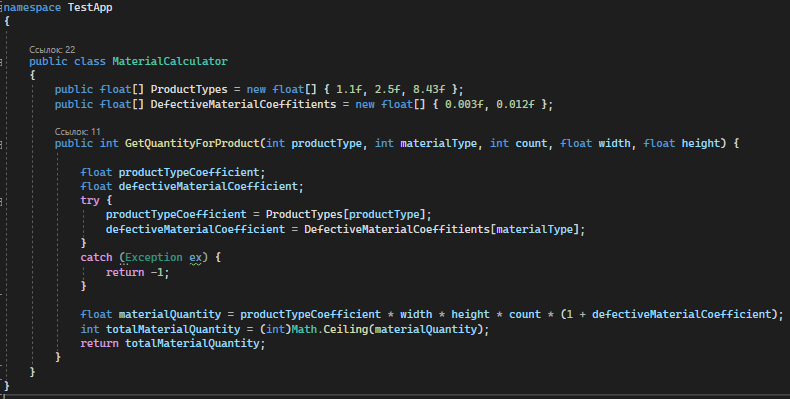


Рисунок 2 – Программный код разработанной функциональности

Для проверки правильной работы разработанной функции были создан проект с модульными тестами в количестве 10 штук.

Результат выполнения тестов продемонстрирован на рисунке 3.

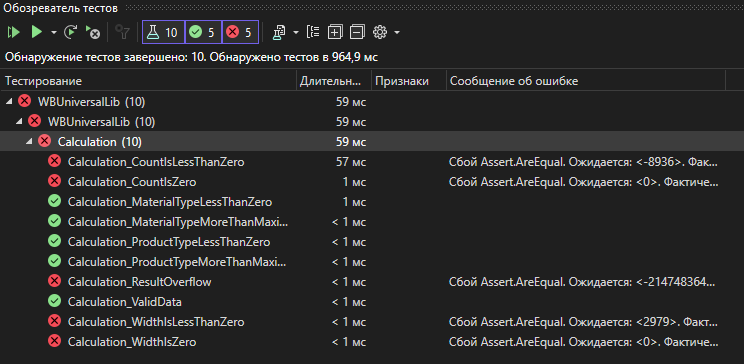


Рисунок 3 – Результат выполнения тестов

В ходе выполнения тестов выявились ошибки в работе функции. Программный код функции был исправлен, новый код продемонстрирован на рисунке 4.

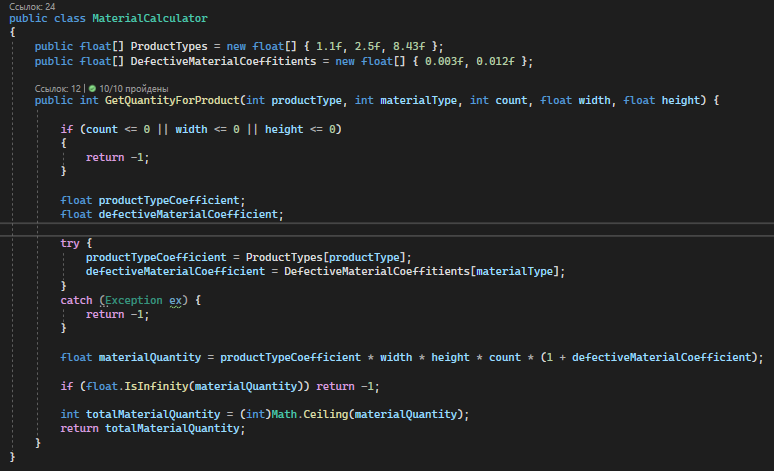


Рисунок 4 – Измененный программный код функции

После изменения программного кода функции был проведен повторный запуск модульных тестов. Результат перезапуска модульных тестов продемонстрирован на рисунке 5.

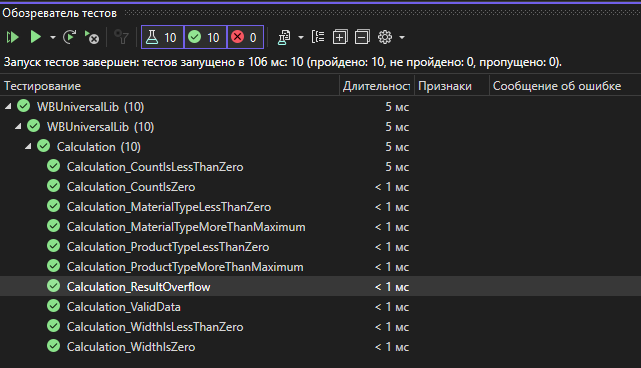


Рисунок 5 – Результат выполнения тестов

Также были созданы 5 Test-Case для тестирования разработанного метода. Test-case приложены ниже.

Аннотация теста:

|  |  |
| --- | --- |
| **Название проекта** | TestApp |
| **Рабочая версия** | 1.00 |
| **Имя тестирующего** | Баланин Э.Н. |
| **Дата(ы) теста** | 01.03.2023 |

**Таблица 1 – Тестовый случай 1**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | TestApp |
| **Рабочая версия** | 1.00 |
| **Имя тестирующего** | Баланин Э.Н. |
| **Дата(ы) теста** | 01.03.2023 |
| **Тестовый пример #** | TC\_OVRFLW\_1 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Средний |
| **Заголовок/название теста** | Проверка работоспособности при слишком больших значениях |
| **Краткое изложение теста** | Тест должен проверить правильно ли функционирует метод при слишком больших значениях |
| **Этапы теста** | В поле «количество» ввести 20  В поле «ширина» ввести число «340282300000000000000000000000000000000»  В поле «высота» ввести «30»  В поле «Номер типа продукта» ввести 1  В поле «Номер типа материала» ввести 1 |
| **Тестовые данные** | Ширина - 340282300000000000000000000000000000000  Высота – 30  Номер типа продукта – 1  Номер типа материала – 1  Количество - 20 |
| **Ожидаемый результат** | -1 |
| **Фактический результат** | -1 |
| **Предварительное условие** | Запущен файл CalculatorForm.exe |
| **Постусловие** |  |
| **Статус** *(Зачет/Незачет)* | Зачет |
| **Примечания/комментарии** |  |

**Таблица 2 – Тестовый случай 2**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | TestApp |
| **Рабочая версия** | 1.00 |
| **Имя тестирующего** | Баланин Э.Н. |
| **Дата(ы) теста** | 01.03.2023 |
| **Тестовый пример #** | TC\_NEGATIVE\_1 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Средний |
| **Заголовок/название теста** | Проверка работоспособности при отрицательном значении ширины |
| **Краткое изложение теста** | Тест должен проверить правильно ли функционирует метод при отрицательном значении ширины |
| **Этапы теста** | В поле «количество» ввести 20  В поле «ширина» ввести число «-854»  В поле «высота» ввести «30»  В поле «Номер типа продукта» ввести 1  В поле «Номер типа материала» ввести 1 |
| **Тестовые данные** | Ширина: -854  Высота: 30  Номер типа продукта: 1  Номер типа материала: 1  Количество: 20 |
| **Ожидаемый результат** | -1 |
| **Фактический результат** | -1 |
| **Предварительное условие** | Запущен файл CalculatorForm.exe |
| **Постусловие** |  |
| **Статус** *(Зачет/Незачет)* | Зачет |
| **Примечания/комментарии** |  |

**Таблица 3 – Тестовый случай 3**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | TestApp |
| **Рабочая версия** | 1.00 |
| **Имя тестирующего** | Баланин Э.Н. |
| **Дата(ы) теста** | 01.03.2023 |
| **Тестовый пример #** | TC\_NEGATIVE\_2 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Средний |
| **Заголовок/название теста** | Проверка работоспособности при отрицательном значении высоты |
| **Краткое изложение теста** | Тест должен проверить правильно ли функционирует метод при отрицательном значении высоты |
| **Этапы теста** | В поле «количество» ввести 20  В поле «ширина» ввести число «30»  В поле «высота» ввести «-854»  В поле «Номер типа продукта» ввести 1  В поле «Номер типа материала» ввести 1 |
| **Тестовые данные** | Ширина: -854  Высота: 30  Номер типа продукта: 1  Номер типа материала: 1  Количество: 20 |
| **Ожидаемый результат** | -1 |
| **Фактический результат** | -1 |
| **Предварительное условие** | Запущен файл CalculatorForm.exe |
| **Постусловие** |  |
| **Статус** *(Зачет/Незачет)* | Зачет |
| **Примечания/комментарии** |  |

**Таблица 4 – Тестовый случай 4**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | TestApp |
| **Рабочая версия** | 1.00 |
| **Имя тестирующего** | Баланин Э.Н. |
| **Дата(ы) теста** | 01.03.2023 |
| **Тестовый пример #** | TC\_ROUND\_2 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Высокий |
| **Заголовок/название теста** | Проверка округления итогового числа |
| **Краткое изложение теста** | Тест должен проверить правильно округляет ли вверх функция итоговое число |
| **Этапы теста** | В поле «количество» ввести «15»  В поле «ширина» ввести число «45»  В поле «высота» ввести «20»  В поле «Номер типа продукта» ввести 2  В поле «Номер типа материала» ввести 0 |
| **Тестовые данные** | Ширина: 45  Высота: 20  Номер типа продукта: 2  Номер типа материала: 0  Количество: 15 |
| **Ожидаемый результат** | 114147 |
| **Фактический результат** | 114147 |
| **Предварительное условие** | Запущен файл CalculatorForm.exe |
| **Постусловие** |  |
| **Статус** *(Зачет/Незачет)* | Зачет |
| **Примечания/комментарии** |  |

**Таблица 5 – Тестовый случай 5**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | TestApp |
| **Рабочая версия** | 1.00 |
| **Имя тестирующего** | Баланин Э.Н. |
| **Дата(ы) теста** | 01.03.2023 |
| **Тестовый пример #** | TC\_PROD\_TYPE\_NOT\_EXIST\_1 |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Высокий |
| **Заголовок/название теста** | Проверка на функционировании при неверном значении типа продукта |
| **Краткое изложение теста** | Тест должен проверить правильно ли функционирует метод при передачи слишком большого значения номера типа продукта |
| **Этапы теста** | В поле «количество» ввести «15»  В поле «ширина» ввести число «45»  В поле «высота» ввести «20»  В поле «Номер типа продукта» ввести 8  В поле «Номер типа материала» ввести 0 |
| **Тестовые данные** | Ширина: 45  Высота: 20  Номер типа продукта: 8  Номер типа материала: 0  Количество: 15 |
| **Ожидаемый результат** | -1 |
| **Фактический результат** | -1 |
| **Предварительное условие** | Запущен файл CalculatorForm.exe |
| **Постусловие** |  |
| **Статус** *(Зачет/Незачет)* | Зачет |
| **Примечания/комментарии** |  |

* 1. **Практическая работа №2. Модульное тестирование модуля авторизации магазина компьютерных комплектующих**

# **УП 01.03 РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ**



## Анализ предметной области

Приложение для учета финансов ‑ это приложение, которое помогает пользователям отслеживать свои доходы и расходы, что включает в себя следующие функции:

* бюджетирование: Приложение позволяет отслеживать текущее состояние бюджета;
* категоризация расходов: пользователь может классифицировать свои расходы по категориям, таким как еда, развлечения, транспорт и т.д. Это помогает пользователям понимать, на что они тратят больше всего денег, и управлять своими расходами;
* мониторинг доходов и расходов: приложение помогает пользователю отслеживать все свои доходы и расходы. Это включает в себя функцию ввода транзакций;
* персонализация: приложение может позволять пользователям настроить свои категории расходов и доходов.

## Создание макетов приложения

В окне авторизации для ввода доступны поля «Логин» и поле «Пароль», которые необходимы для ввода одноименных значений. При нажатии на кнопку «Регистрация» в окне авторизации произойдет перенаправление на окно регистрации. При нажатии на кнопку «Зарегистрироваться» в окне регистрации, случае если регистрация на стороне сервера произойдет успешно, произойдет перенаправление обратно в окно авторизации с заполненными полями «Логин» и «Пароль» для дальнейшего входа.

При авторизации с поднятым флагом «Запомнить меня» в случае успешной авторизации данные пользователя запишутся в локальную базу данных, чтобы при повторном запуске программы пользователь не проходил повторно эта авторизации и не создавался новый токен.

На рисунке 1 продемонстрированы макеты авторизации и регистрации.

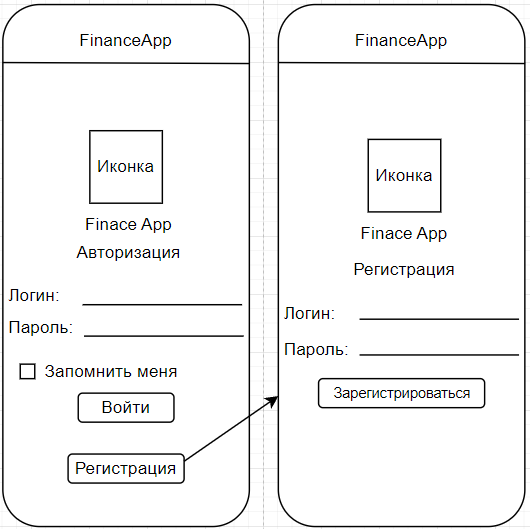


Рисунок 1 – Макет авторизации и регистрации

На рисунке 2 изображены макеты активностей, содержащих в себе весь функционал работы с категориями трат: Отображение списка всех категорий, добавление новой категории, изменение информации о категории и удаление категории. Удаление категории происходит путем нажатия на кнопку «–» около соответствующей категории.

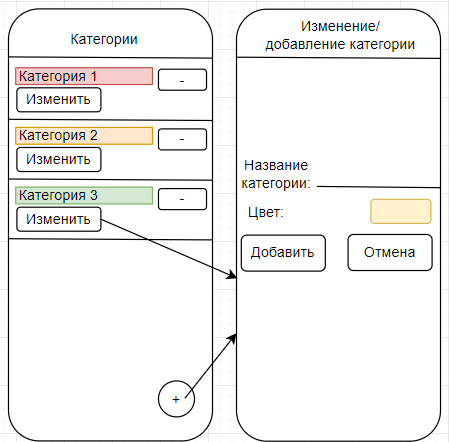


Рисунок 2 – Макет просмотра списка категорий и макет добавления/изменения информации категории

На рисунке 3 изображены макеты списка всех статей расходов, а также макет активности создания новой статьи расхода. Переход на активность создания новой статьи расхода осуществляется при помощи нажатия на кнопку «+». Удаление статьи расхода осуществляется нажатием на интересующем пункте расходов кнопку «-».

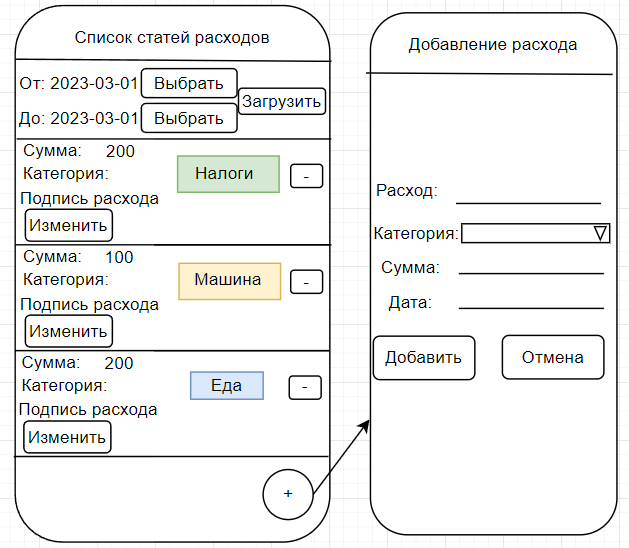


Рисунок 3 – Макет активности отображения расходов и добавления расходов

На рисунке 4 изображена активность изменения информации о существующей статье расхода.

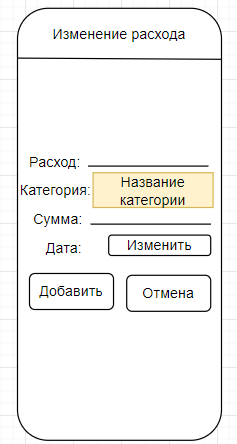


Рисунок 4 – Макет активности изменения расхода

На рисунке 5 продемонстрирован макет окно диаграммы загруженных с сервера расходов разбитых по определенным категориям.

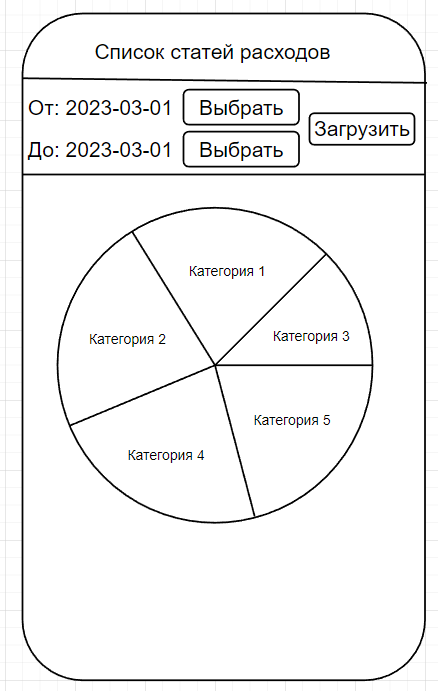


Рисунок 5 – Макет окна с диаграммой расходов по категориям

## Описание функций API

API (Application Programming Interface) – это набор стандартов, протоколов и инструментов, которые используются для взаимодействия между различными программами и приложениями. API позволяет различным приложениям обмениваться данными и функциональностью, необходимой для выполнения определенных задач.

API может представлять собой программный интерфейс, который позволяет разработчикам создавать приложения, которые работают вместе с другими приложениями или сервисами. Например, API социальных сетей позволяют разработчикам создавать приложения, которые используют данные социальных сетей, такие как профили пользователей и новости.

API может быть как публичным, доступным для использования всеми, так и частным, доступным только для определенных пользователей или организаций. Обычно API доступны через интернет, и разработчики используют HTTP-запросы для отправки и получения данных.

API могут использоваться для создания различных приложений и сервисов, таких как мобильные приложения, веб-приложения, социальные сети, электронные магазины и т.д. Разработчики могут использовать API для интеграции своих приложений с другими приложениями или сервисами, что помогает увеличить функциональность и улучшить пользовательский опыт.

В целом, API является ключевым инструментом для современной разработки приложений, поскольку он позволяет создавать приложения, которые легко интегрируются с другими системами и сервисами, а также используют общие стандарты и протоколы для обмена данными.

API приложения учета финансов предоставляет интерфейс для взаимодействия с системой учета финансов. API интерфейс помогает пользователям отслеживать свои финансовые операции, создавать кошельки, записывать транзакции, анализировать расходы и доходы, а также управлять своими финансами. Предоставленный API предоставляет набор методов для создания, чтения, обновления и удаления кошельков, транзакций и категорий расходов. Каждая транзакция имеет свою категорию, что помогает пользователям быстро определить, на что были потрачены деньги.

Пользователи могут использовать API для получения отчетов и статистики по своим доходам и расходам, что помогает им понимать, куда уходят их деньги. Кроме того, API обеспечивает безопасность данных пользователей, используя авторизацию и аутентификацию.

Использование API учета финансов позволяет разработчикам интегрировать приложение с другими системами и сервисами, а также создавать пользовательские интерфейсы для взаимодействия с приложением.

Далее будет описан основной функционал предоставляемый API для приложения учета финансов.

Для правильного функционирования всех методов предоставляемого API необходимо при обращении к методу дописывать в начале следующий адрес до метода: «http://spbcoit.ru:80/lab/budget/api».

API предоставляет возможность авторизации пользователя. На рисунке 6 продемонстрировано описание API метода авторизации. Для авторизации необходимо отправить ключи «name» и «secret» – логин и пароль соответственно в формате JSON. Функция расположена по адресу «/rpc/sign\_in».

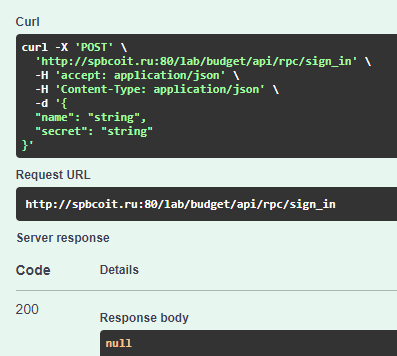


Рисунок 6 – Описание функции «/rpc/sign\_in»

Также API интерфейс предоставляет средства для регистрации нового аккаунта. Для регистрации необходимо отправить JSON объект, содержащий ключи «name» и «secret» - логин и пароль соответственно по адресу «/rpc/register\_account». При успешной регистрации сервер вернет код 200 и объект булевой переменной true. Пример функционирования продемонстрирован на рисунке 7.

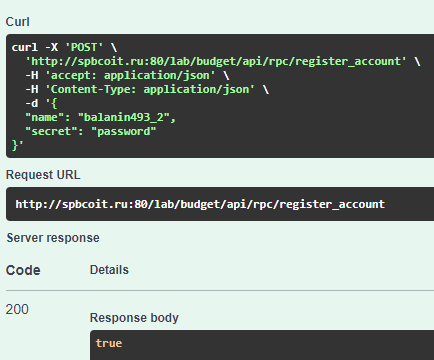


Рисунок 7 – Результат работы функции «/rpc/register\_account»

API интерфейс также предоставляет возможность закрытия текущей сессии (при авторизации пользователь получает уникальный ключ, который является идентификатором текущей сессии пользователя, при помощи данного идентификатора сервер понимает какой пользователь обращается к серверу не передавая каждый раз логин и пароль пользователя.

Для закрытия сессии необходимо воспользоваться функцией «/rpc/sign\_out». На рисунке 8 продемонстрирован пример использования функции «/rpc/\_sign\_out».

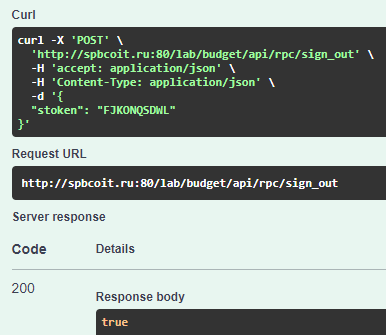


Рисунок 8 – Пример работы функции «/rpc/sign\_out»

Также API поддерживает функцию удаления аккаунта по адресу «/rpc/close\_account». На рисунке 9 продемонстрирован пример работы данного метода.

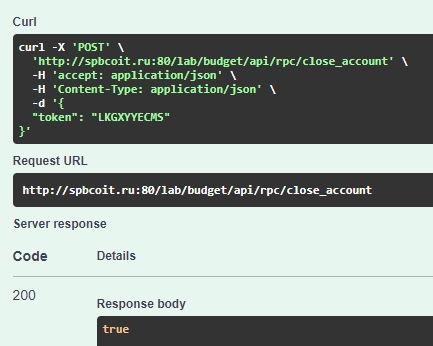


Рисунок 9 – Пример работы функции «/rpc/close\_account»

Также API интерфейс предоставляет функцию для создания новой категории (далее для каждой статьи расхода будет указываться созданная категория). Пример работы функции «/rpc/add\_category» продемонстрирован на рисунке 10.

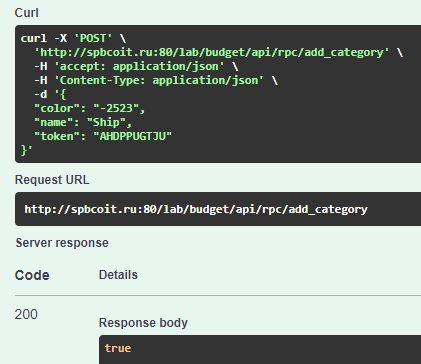


Рисунок 10 – Пример работы функции «/rpc/add\_category»

API интерфейс предоставляет функцию для просмотра списка всех категорий. Пример работы функции «/rpc/get\_categories» продемонстрирован на рисунке 11.

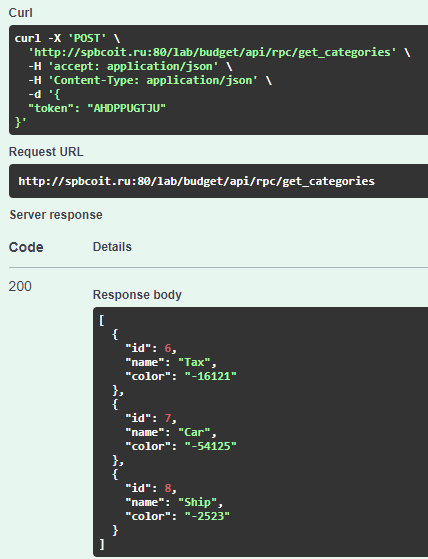


Рисунок 11 – Пример работы функции «/rpc/get\_categories»

На рисунке 12 продемонстрирован пример работы функции «/rpc/update\_category», предоставляющей возможность изменения характеристик существующей категории.

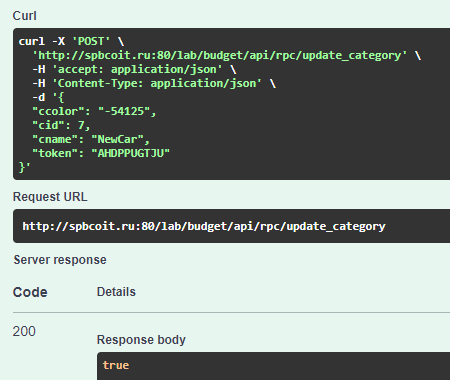


Рисунок 12 – Пример работы функции «/rpc/update\_category»

На рисунке 13 продемонстрирован пример работы функции «/rpc/delete\_category», позволяющей удалять ранее созданную категорию.

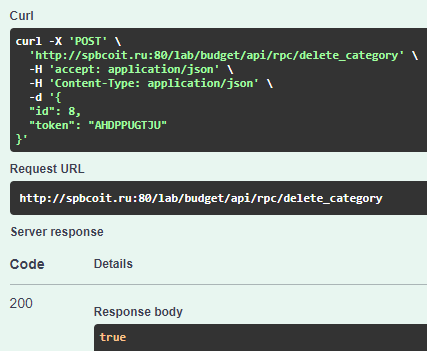


Рисунок 13 – Пример работы функции «/rpc/delete\_category»

Также API интерфейс предоставляет функцию для создания новой статьи расходов. Пример работы функции «/rpc/add\_expense» продемонстрирован на рисунке 14.

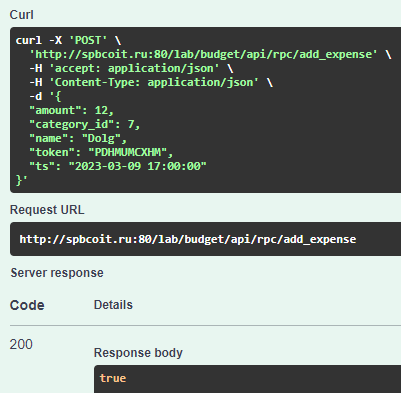


Рисунок 14 – Пример работы функции «/rpc/add\_expense»

API интерфейс предоставляет функцию для просмотра списка всех статей расходов. Пример работы функции «/rpc/get\_expenses» продемонстрирован на рисунке 15.

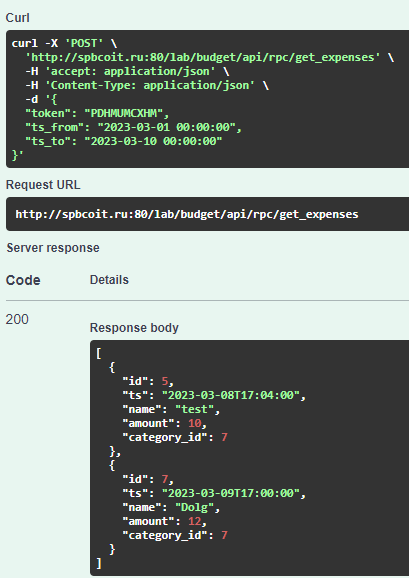


Рисунок 15 – Пример работы функции «/rpc/get\_expenses»

На рисунке 16 продемонстрирован пример работы функции «/rpc/update\_category», предоставляющей возможность изменения характеристик существующей категории.

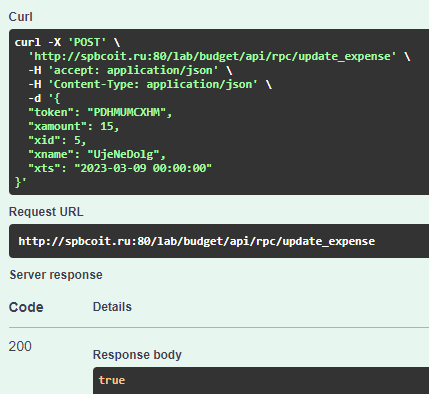


Рисунок 16 – Пример работы функции «/rpc/update\_category»

На рисунке 17 продемонстрирован пример работы функции «/rpc/delete\_category», позволяющей удалять ранее созданную категорию.

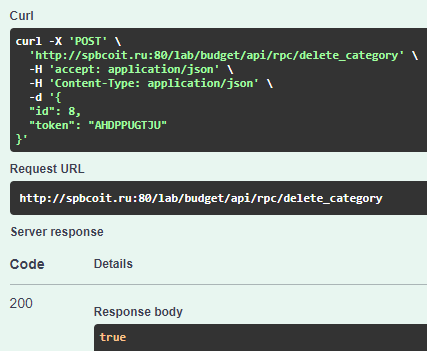


Рисунок 17 – Пример работы функции «/rpc/delete\_expense»

## Интерфейс пользователя

В данном разделе отчета представлена схема переходов интерфейса пользователя а также сам интерфейс пользователя и его описание. Данная схема демонстрирует наглядно в каком порядке происходит вызов и переключение окон приложения (правильный порядок вызова окон). На рисунке 18 продемонстрирована схема переходов интерфейса пользователя.

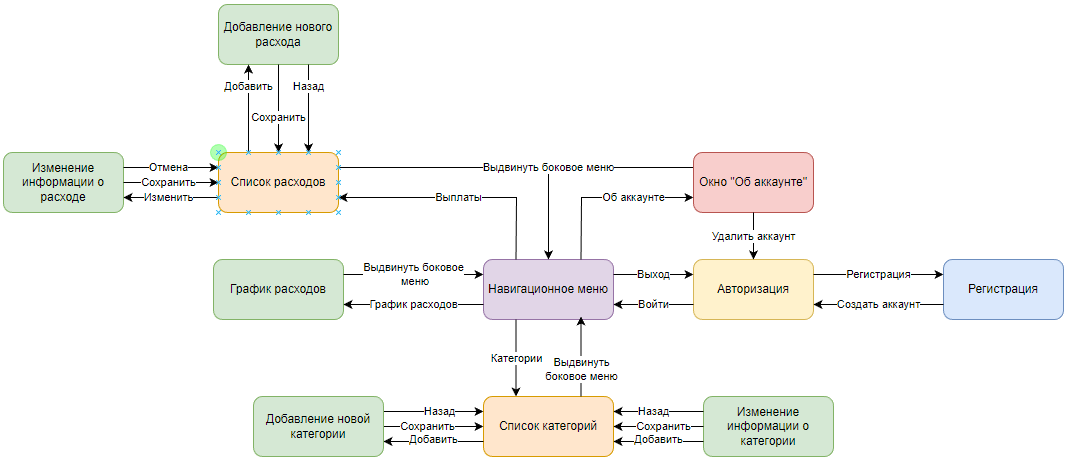


Рисунок 18 – Схема переходов окон интерфейса пользователя

**Авторизация пользователя**

На рисунке продемонстрирован внешний вид окна авторизации пользователя

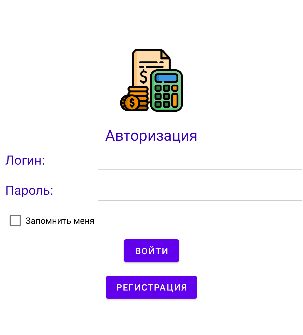


Рисунок 19 – Внешний вид окна авторизации

По нажатию на кнопку «Регистрация» происходит переход на окно регистрации. По нажатию на кнопку «Войти» при правильно введенном логине и пароле произойдет переход в основное навигационное меню.

**Регистрация пользователя**

На рисунке 20 продемонстрирован внешний вид окна регистрации нового пользователя. По нажатию на кнопку «Зарегистрироваться» произойдет переход на окно авторизации (при условии что поля «Логин» и «Пароль» заполнены).

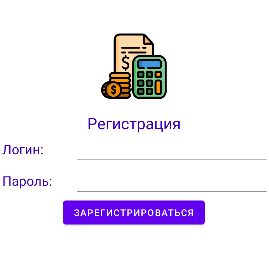


Рисунок 20 – Внешний вид окна регистрации

**Окно списка расходов**

На рисунке 21продемонстрирован внешний вид окна списка расходов содержащего в себе несколько расходов.



Рисунок 21 – Внешний вид окна списка расходов

**Окно создания расхода/изменения информации о расходе**

На рисунке 22 продемонстрировано окно создания нового расхода а также окно изменения существующего расхода. Для перехода на окно изменения существующего расхода необходимо нажать на кнопку «Изменить» в окне списка расходов, продемонстрированном на рисунке 21. Для перехода на окно создания расхода необходимо нажать на кнопку «+» в окне списка расходов, продемонстрированном на рисунке 21.

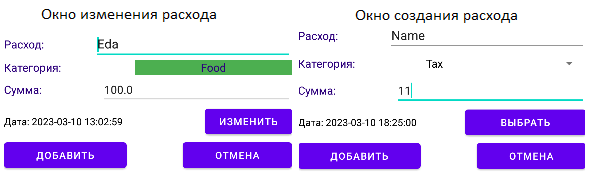


Рисунок 22 – Внешний вид окон изменения расхода и создания нового расхода

**Окно списка категорий**

На рисунке 23 продемонстрирован внешний вид окна списка категорий. По нажатию на кнопку «Изменить произойдет переход на окно изменения информации о категории.



Рисунок 23 – Внешний вид окна списка категорий

**Окно создания категории/изменения информации о категории**

Окно создания новой категории и окно изменения информации о существующей категории внешне ничем не отличаются, внешний вид этих окон продемонстрирован на рисунке 24. Чтобы открыть окно создания категории необходимо нажать на кнопку «+» в окне списка всех категорий, продемонстрированном на рисунке 23.

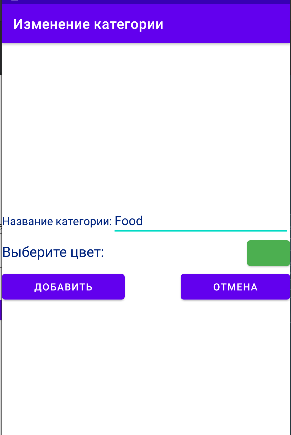


Рисунок 24 – Внешний вид окна добавления/изменения категории

**Окно диаграммы расходов**

На рисунке 25 продемонстрирован внешний вид окна демонстрации диаграммы расходов за определенный период по категориям.

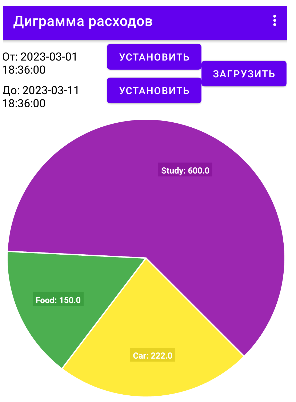


Рисунок 25 – Внешний вид окна диаграммы расходов

## Руководство пользователя

При запуске приложения первое что видит пользователь – окно авторизации. Необходимо ввести логин и пароль для авторизации пользователя. На рисунке 26 изображено меню авторизации приложения.

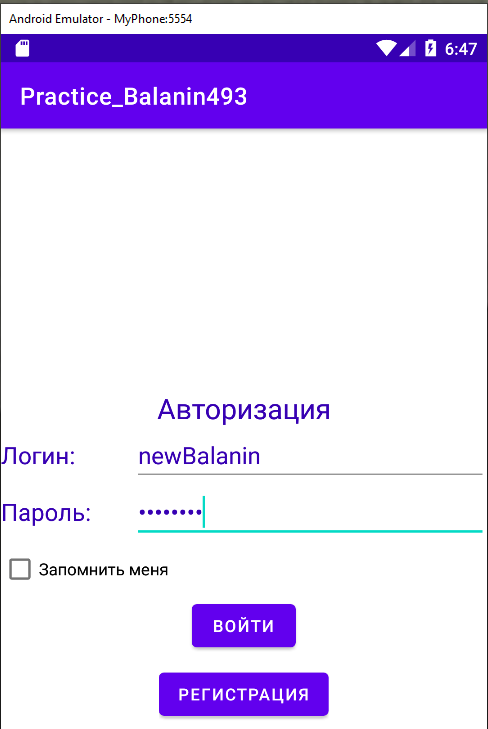


Рисунок 26 – Окно авторизации

Если нажать на кнопку регистрация, отроется окно продемонстрированное на рисунке 27.

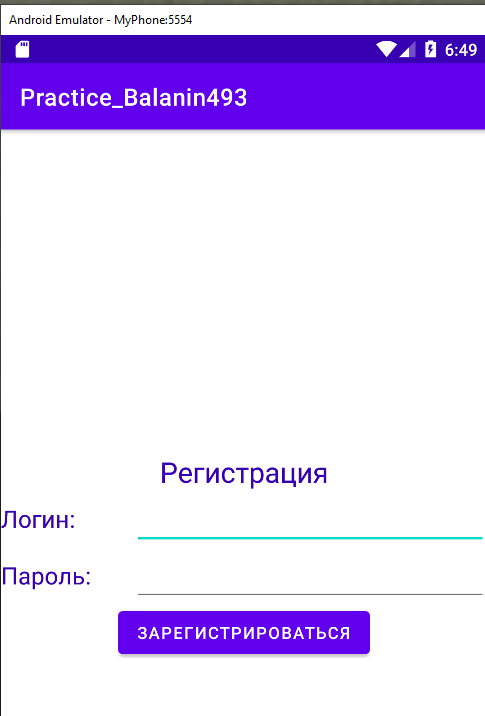


Рисунок 27 – Окно регистрации

Чтобы зарегистрироваться введите в поле «Логин» и пароль ваш логин и пароль для регистрации нового аккаунта и нажмите кнопку «Зарегистрироваться», в случае если пользователя с таким логином не существует , то произойдет автоматическое перенаправление на окно авторизации с заполненными значениями «Логин» и «Пароль».

При успешной авторизации следующим окном будет окно расходов, продемонстрированное на рисунке 28. Здесь необходимо установить дату начала и конца для выборки данных из сервера. Установить даты можно нажатием на кнопку «Установить». При нажатии на эту кнопку появится окно для выбора даты продемонстрированное на рисунке 28. После выбора дат «От» и «До» необходимо нажать на кнопку «Загрузить» для загрузки расходов из сервера. Список загруженных расходов продемонстрирован на рисунке 28. При отображении расхода название категории будет подсвечено соответствующем цветом данной категории.

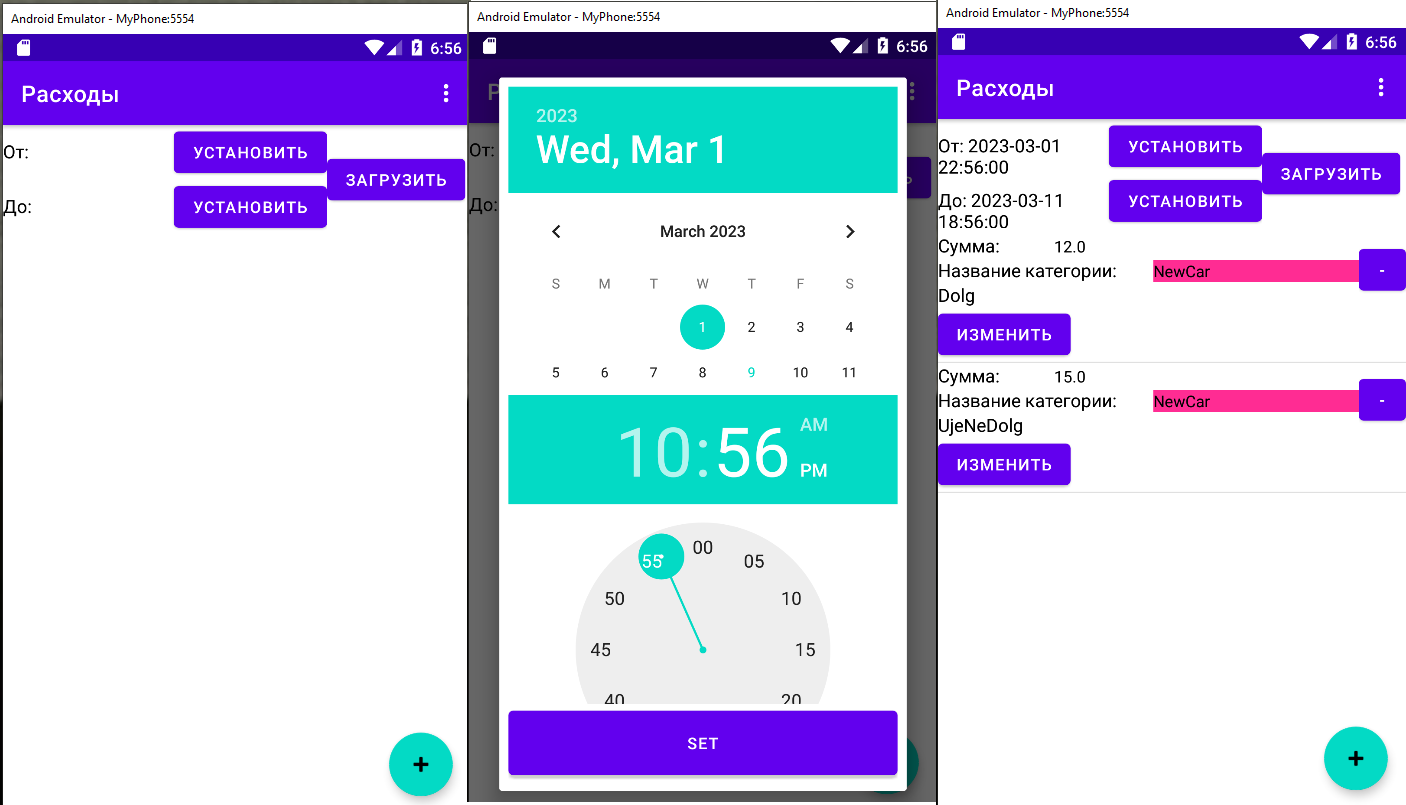


Рисунок 28 – Окно расходов, окно выбора даты и отображение выбранных расходов

Для изменения информации отображаемой статьи расхода необходимо нажать на кнопку «Изменить» у соответствующей статьи расхода. При нажатии на кнопку произойдет переход на окно изменения информации выбранного расхода. На рисунке 29 изображена данная форма.

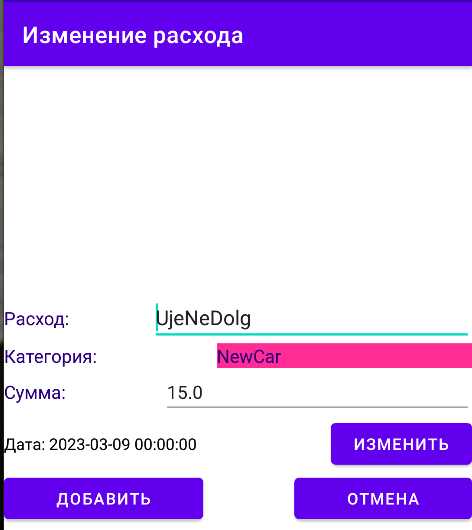


Рисунок 29 – Окно изменения информации о расходе

На рисунке 30 продемонстрирован результат изменения статьи расхода.

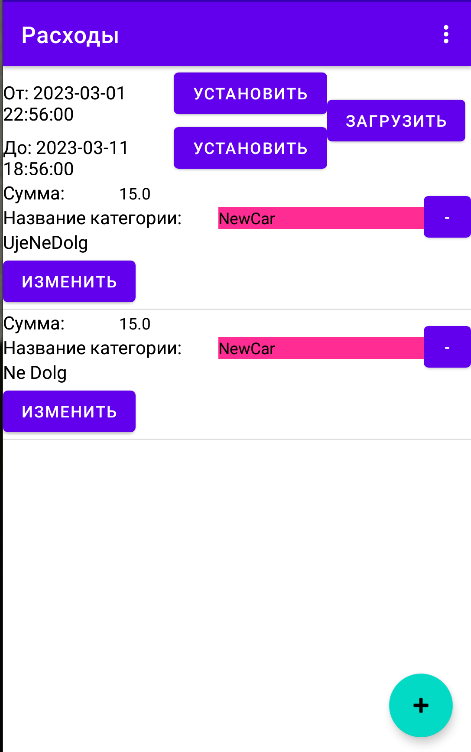


Рисунок 30 – Результат изменения информации о расходе

Чтобы удалить необходимую статью расхода, нужно нажать на кнопку «-» напротив данной статьи расхода. При этом появится окно подтверждения действия. Окно подтверждения действия и результат удаления продемонстрированы на рисунке 31.

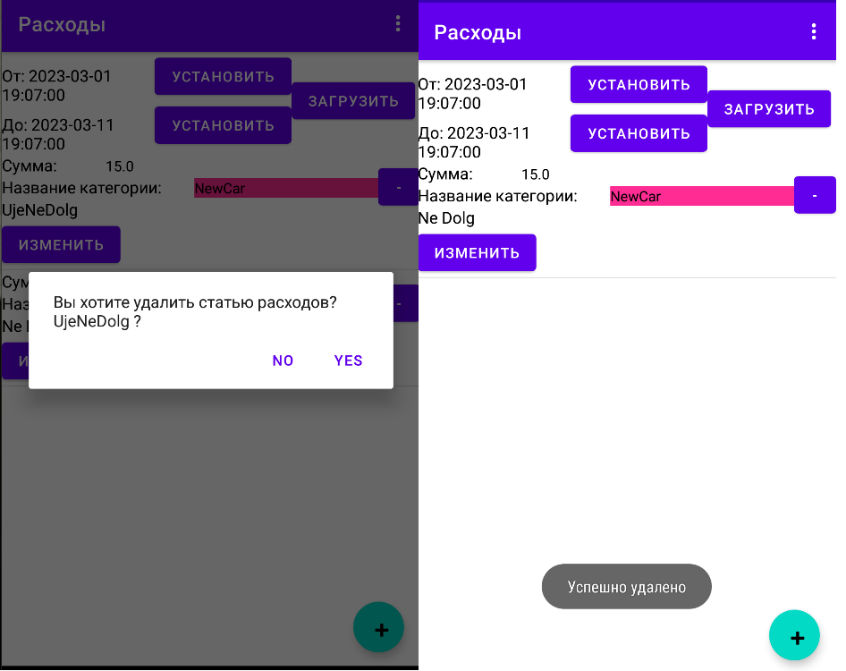


Рисунок 31 – Окно подтверждения удаления расхода и результат удаления

Для создания нового расхода нужно нажать на кнопку «+». Откроется окно для создания расхода. На рисунке 32 продемонстрировано окно создания расхода и результат его добавления.

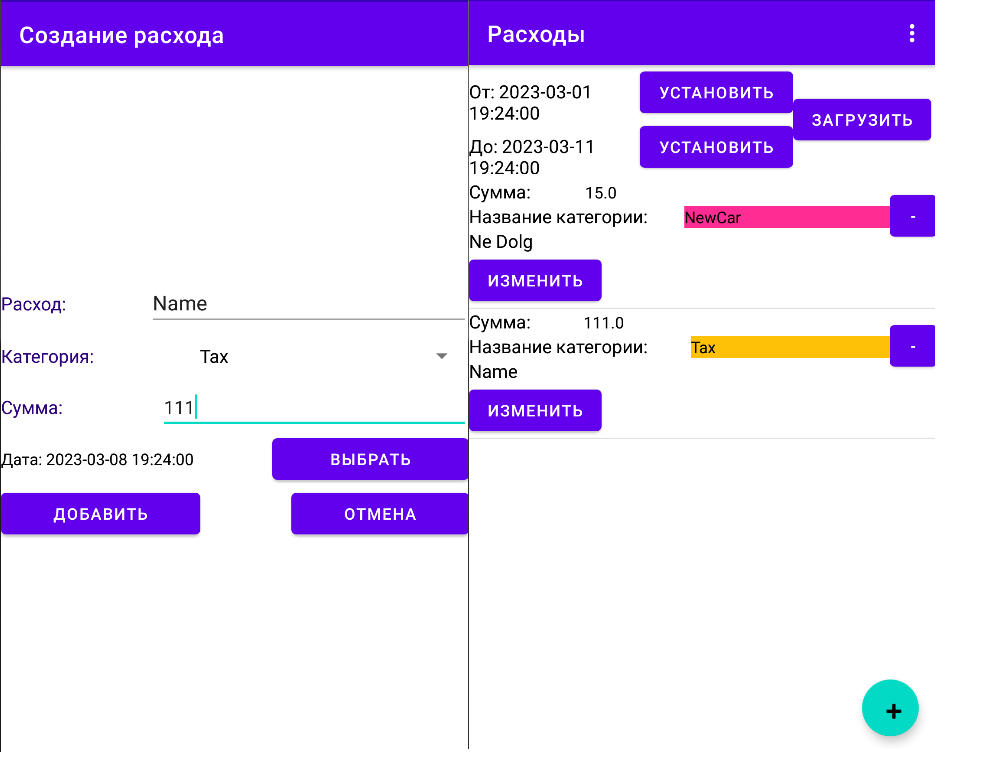


Рисунок 32 – Окно добавления расхода и результат его добавления

Для перехода на вкладку «Категории» используется боковое выдвижное меню. Чтобы оно отобразилось необходимо провести пальцем с левом части экрана в правую сторону. Меню для навигации продемонстрировано на рисунке 33.

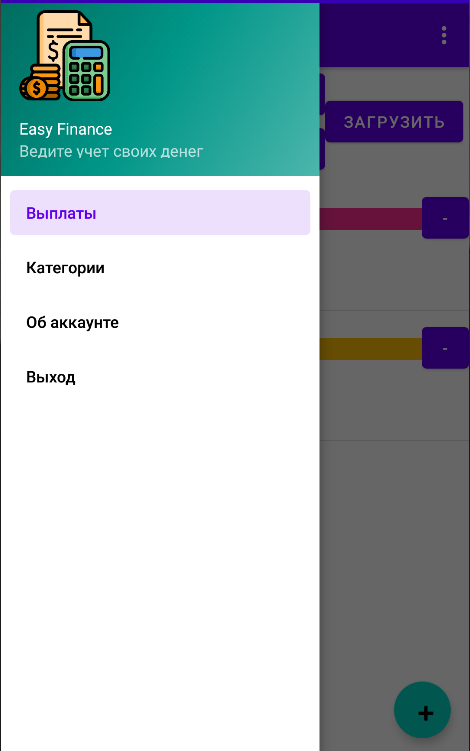


Рисунок 33 – Меню навигации приложения

При нажатии на пункт «Категории» происходит переход на окно категорий, продемонстрированное на рисунке 34.

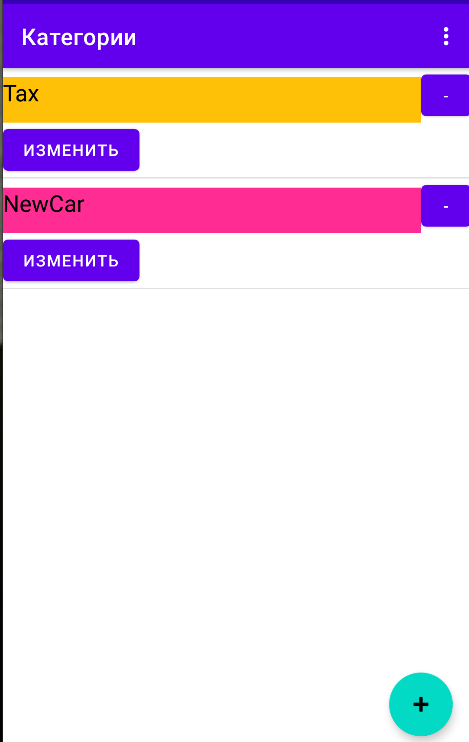


Рисунок 34 – Окно просмотра категорий

В данном окне отображается список всех существующих на сервере записей категорий. Для добавления новой категории необходимо нажать на кнопку «+». На рисунке 35 продемонстрировано окно добавления новой категории. При нажатии на кнопку «Выбрать цвет» отображается окно для выбора цветов, продемонстрированное на рисунке 35. Также на рисунке 35 продемонстрирован результат добавления новой категории.

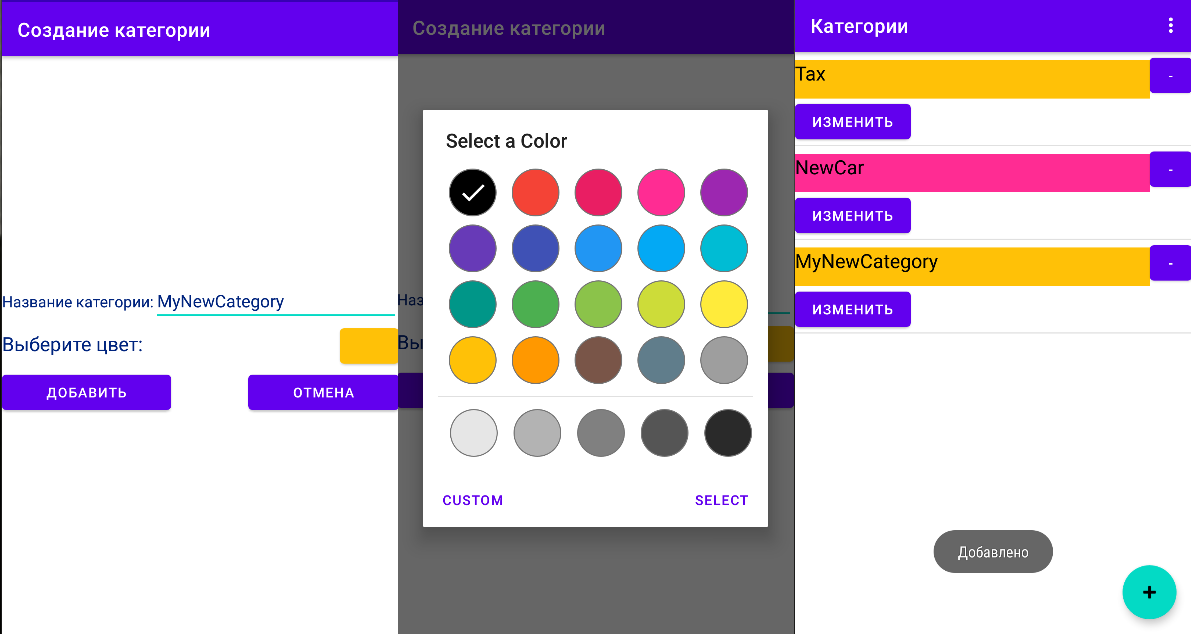


Рисунок 35 – Окна добавления категории, выбора цвета и отображение изменений

При нажатии на кнопку «Изменить» у определенной категории отобразится окно изменений, продемонстрированное на рисунке 36.

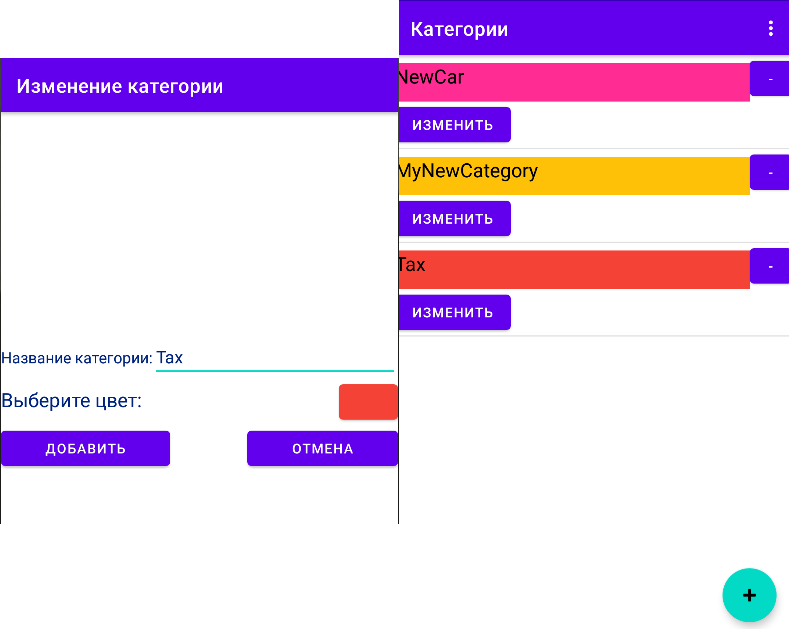


Рисунок 36 – Изменение категории и демонстрация результата

Для удаления выбранной категории необходимо нажать на кнопку «-». При удалении категории вылезет подтверждающее сообщение. На рисунке 37 продемонстрирован результат удаления категории и окно подтверждения результата.

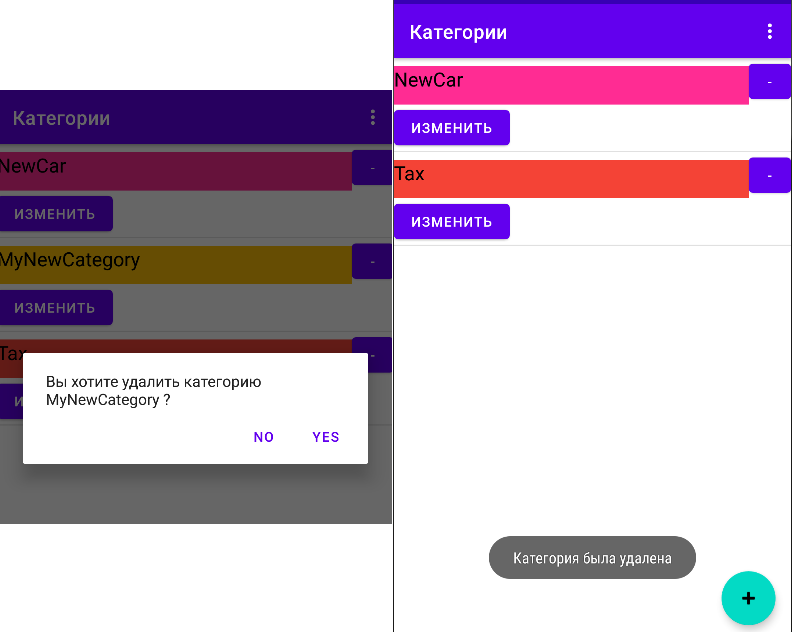


Рисунок 37 – Удаление категории

В меню навигации можно перейти в пункт «Об аккаунте». При нажатии на кнопку «Удалить аккаунт» вылезает уведомление о подтверждении действия. Результат удаления аккаунта и окно «Об аккаунте» продемонстрирован на рисунке 38.

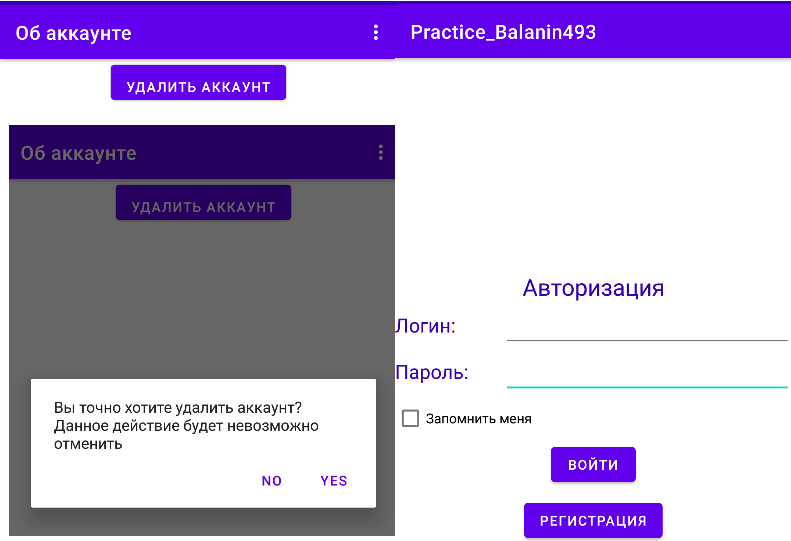


Рисунок 38 – Окно «Об аккаунте», удаление аккаунта и результат удаления