Министерство НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Курский государственный университет»

Факультет физики, математики, информатики

Кафедра программного обеспечения и администрирования информационных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(бакалаврская работа)

на тему: «ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА КЛАСТЕРИЗАЦИИ ДЛЯ ПРОДУКТОВОГО МАГАЗИНА»

Обучающегося 4 курса

очной формы обучения

направления подготовки

02.03.03 Математическое

обеспечение и администрирование информационных систем

Направленность (профиль) Проектирование информационных систем и баз данных

Мвеемба Элиас Мунсанда

Руководитель:

д.п.н., профессор Кудинов В.А.

Консультанты:

доцент Демченко Алина Алексеевна,

доцент Соколова Ирина Александровна

Допустить к защите:

и.о. заведующего кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Макаров К.С./

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.

Курск, 2021

# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc73523937)

[1 Анализ требований к информационной системе 6](#_Toc73523938)

[1.1 Описание и анализ предметной области 6](#_Toc73523939)

[1.1.1 Алгоритма K-means кластеризации 9](#_Toc73523940)

[1.1.2 Алгоритма K-means кластеризации основан на анализ RFM 10](#_Toc73523941)

[1.2 Обзор и анализ возможных альтернатив 13](#_Toc73523942)

[1.3 Анализ функциональных и эксплуатационных требований 13](#_Toc73523943)

[1.3.1 Стандарты 13](#_Toc73523944)

[1.3.2 Функциональные требования пользователя 14](#_Toc73523945)

[1.3.3 Входные данные 14](#_Toc73523946)

[1.3.4 Выходные данные 14](#_Toc73523947)

[1.3.5 Требования к интерфейсу 14](#_Toc73523948)

[1.3.6 Требования к надежности 15](#_Toc73523949)

[1.3.7 Требования к программной документации 15](#_Toc73523950)

[1.3.9 Модель вариантов использования 16](#_Toc73523951)

[1.3.9.1 Диаграмма вариантов использования 16](#_Toc73523952)

[1.3.9.2 Описание варианта использования «Выполнить кластеризацию» 17](#_Toc73523953)

[2 Проектирование информационной системы 19](#_Toc73523954)

[2.1 Разработка архитектуры системы 19](#_Toc73523955)

[2.2 Разработка модели предметной области 19](#_Toc73523956)

[2.3 Разработка алгоритма функционирования системы 25](#_Toc73523957)

[2.4 Проектирование интерфейса пользователя 28](#_Toc73523958)

[2.5 Реляционная модель данных 28](#_Toc73523959)

[2.6. Проектирование классов предметной области 29](#_Toc73523960)

[2.6.1. Построение диаграмм последовательностей для варианта использования «Выполнить кластеризацию» 29](#_Toc73523961)

[2.6.2 Построение диаграммы кооперации 29](#_Toc73523962)

[3 Реализация системы 31](#_Toc73523963)

[3.1 Реализация программного обеспечения системы 31](#_Toc73523964)

[3.1.1 Разработка диаграммы компонентов 31](#_Toc73523965)

[3.1.2 Объекты интерфейса пользователя 31](#_Toc73523966)

[3.2 Реализация технического обеспечения 33](#_Toc73523967)

[4 Анализ результатов 33](#_Toc73523968)

[4.1 Разработка тестов и тестирование системы 33](#_Toc73523969)

[4.2 Анализ эргономических эффектов от использования системы 36](#_Toc73523970)

[ЗАКЛЮЧЕНИЕ 37](#_Toc73523971)

**ВВЕДЕНИЕ**

Разрабатываемый программный продукт является системой кластеризации пользователей магазина в целях определения их предпочтений и формирования рекомендаций по рекламе.

Кластеризация – это задача разделения совокупности или точек данных на несколько групп таким образом, чтобы совокупности данных в группах были более похожи на другие точки данных в совокупности данных в группе и отличались от точек данных в других группах. По сути, это набор объектов на основе сходства и несходства между ними. Кластеризация очень важна, поскольку она определяет внутреннюю группировку среди имеющихся немаркированных данных. Из-за вышеупомянутой важности его можно использовать в интернет-магазине продуктов для группировки клиентов в разные группы в зависимости от различных факторов, таких как:

– какие товары они покупают,

– как часто они покупают эти товары, когда они покупают эти продукты,

– какие продукты они чаще всего просматривают на сайте или в приложении и т. д.

Использование различных собранных групп, основанных на общих свойствах этих групп, дает возможность отображать продукты, которые являются наиболее важными для каждого покупателя, и дает возможность предсказать, что покупатель может купить в какое конкретное время года.

Целью разработки приложений является создание системы кластеризации, которая группирует клиентов определенных интернет-магазинов продуктов в несколько групп и дает предложения о том, какие продукты рекламировать для каждой результирующей группы.

Основными задачами разработки являются:

1. Формирование групп на основе кластеризации.
2. Обеспечение доступа к истории покупок клиентов.
3. Обеспечение просмотра различных кластеризованных групп с целью формирования предложений для рекламы.

Программный продукт предназначен для использования в любом онлайн магазинах.

Для разработки программного продукта применяется среда Sublime text. Вместе с СУБД Microsoft SQL server management studios для создания информационной системы используется CASE-средство Rational Rose Enterprise Edition v2001a.

1. **Анализ требований к информационной системе**

## **1.1 Описание и анализ предметной области**

В настоящее время предлагаемые продукты на сайтах продуктов выбираются случайным образом, этот способ выбора предложений делает пользовательский опыт менее приятным и снижает шансы клиента или пользователя сайта купить столько продуктов, сколько они купили бы, если бы им были предоставлены правильные предлагаемые продукты.

Вышеупомянутые недостатки создают необходимость в системе, которая помогала бы обрабатывать пользовательские данные и предоставлять наилучшие предложения на основе истории пользователей.

Система позволит:

– Предоставление лучших индивидуальных предложений по продукту для каждого отдельного пользователя.

– Обеспечение лучшего пользовательского опыта, потому что клиенты будут иметь дело только с продуктами, которые их интересуют.

– Предоставление важные данные о пользователях менеджеру продуктового магазина, который помогает менеджеру знать, какие продукты следует запастись и когда их запастись.

Система будет использоваться менеджером любого интернет-магазина товаров. При наличии данных о пользователях, таких как предыдущие покупки, система будет фильтровать данные и группировать их в группы со сходством и предоставлять индивидуальные предложения для каждого клиента.

На рисунке 1 представлена контекстная DFD-диаграммы системы сегментации рынка на основе кластеризации в BPWin. Входными данными служит информация покупателей (Market segmentation system), выходными график ( Graph) и таблица (Table) с результатами анализа.

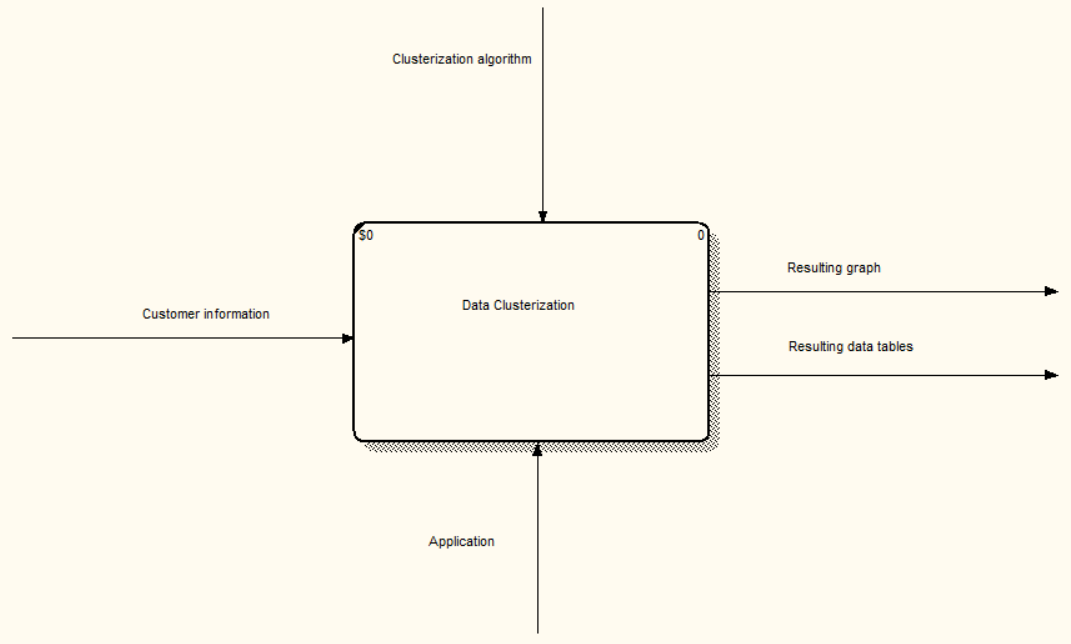


Рисунок 1 – Контекстная SADT-диаграмма системы

На рисунке 2 приведено описание бизнес-процесса в BPMN. На выходе действий: «Загрузить данные покупателей», «Провести анализ данных», «Выполнить кластеризации», «вывести результатов»

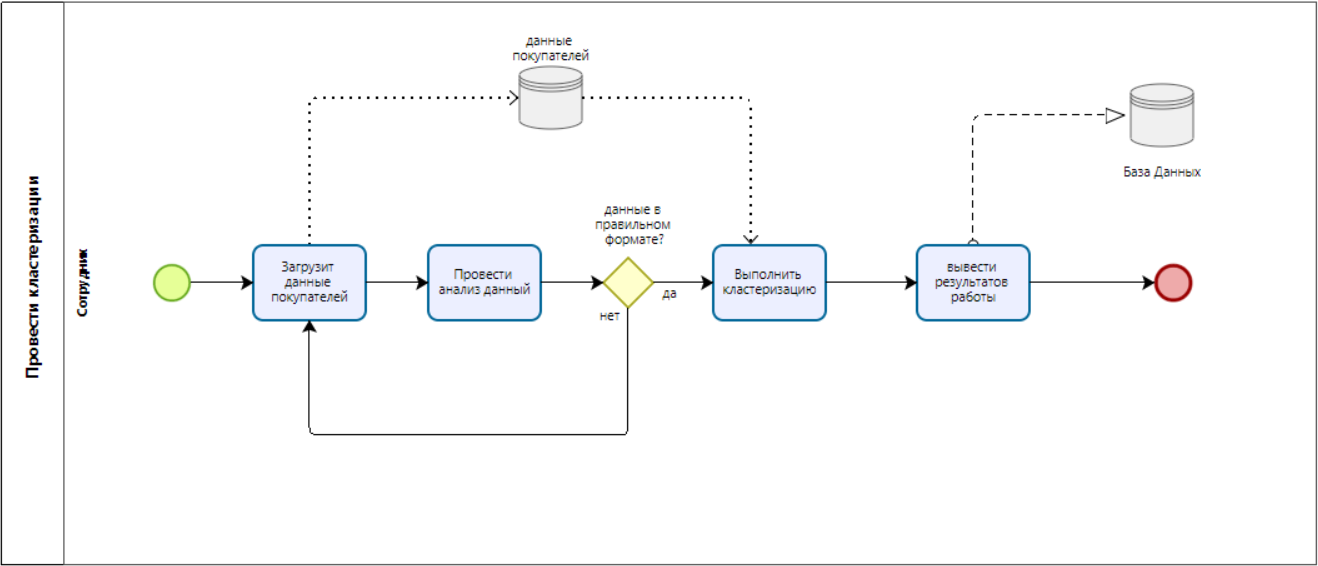


Рисунок 2 – Описание бизнес-процесса «Заключение договора» в BPMN

**1.1.1 Алгоритма K-means кластеризации**

Создаваемая программа будет использовать метод кластеризации K-means.

Алгоритм K-means - это итеративный алгоритм, который пытается разделить набор данных на заранее определенные K отдельных неперекрывающихся подгрупп (кластеров), где каждая точка данных принадлежит только одной группе. Он пытается сделать точки данных внутри кластера как можно более похожими, но при этом сохраняя кластеры как можно более разными (далеко). Он назначает точки данных кластеру таким образом, чтобы сумма квадратов расстояния между точками данных и центроидом кластера (среднее арифметическое всех точек данных, принадлежащих этому кластеру) была минимальной. Чем меньше вариаций внутри кластеров, тем более однородные (похожие) точки данных находятся в одном кластере.

Алгоритм K-means работает следующим образом:

1. Укажите количество кластеров K.
2. Инициализируйте центроиды, сначала перетасовав набор данных, а затем случайным образом выбрав K точек данных для центроидов без замены.
3. Продолжайте итерацию до тех пор, пока центроиды не изменятся, то есть назначение точек данных кластерам не изменится:

– Вычислите сумму квадратов расстояния между точками данных и всеми центроидами.

– Назначьте каждую точку данных ближайшему кластеру (центроиду).

– Вычислите центроиды для кластеров, взяв среднее значение всех точек данных, принадлежащих каждому кластеру.

**1.1.2 Алгоритма K-means кластеризации основан на анализ RFM**

Суть анализа RFM(Недавность, Частота, Денежная стоимость) состоит в том, чтобы разделить клиентов на группы на основе того, как недавно они совершили свою последнюю покупку, как часто они покупают вещи и среднюю стоимость их заказов. Для каждой из этих метрик мы назначаем клиентов в одну из трех групп, которым присваивается номер от 1 до 3.

Недавность:

1 – давние клиенты.

2 – относительно недавние клиенты.

3 – последние клиенты.

Частота:

1 – покупки редко (единичные заказы).

2 – покупки нечасто.

3 – покупки часто.

Денежная стоимость:

1 – низкая стоимость покупок.

2 – средняя стоимость покупок.

3 – высокая стоимость покупок.

Анализ RFM-это просто благо для растущего графика любой компании,

1) в первую очередь потому, что это может помочь в увеличении удержания клиентов, что приведет к увеличению числа откликов -> увеличению коэффициента конверсии -> увеличению дохода для компании.

2)во-вторых, анализ RFM в сочетании с различными другими показателями приоритета может иметь первостепенное значение для поиска моделей поведения Клинтов, что также важно для маркетинга, целей продаж и т. Д.

3) Можно выбрать различные способы маркетинга, и затраты могут быть сокращены за счет огромной маржи, поскольку компании не будут тратить много потерянных дешевых клиентов по сравнению с почти потерянными/потерянными клиентами или начинающими клиентами.

4) Денежные лояльные клиенты могут быть признаны, чтобы побудить компанию к высококачественному, немного высокому ценовому продукту, вместо того, чтобы пугаться и снижать стоимость, приводящую к снижению качества продукции. Благодаря анализу RFM.

5) Вознаграждение лояльных или потенциально лояльных клиентов может быть сделано путем предоставления им бесплатной доставки или предоставления небольших бесплатных услуг или ваучеров, все из-за действительного признания в результате анализа RFM.

Что-то вроде премиальных предложений для крупных платежеспособных клиентов (поскольку не будет беспокойства о больших скидках ) и разумной скидки для высокочастотных средних денежных клиентов, следовательно, эффективно сохраняя обе категории, не тратя много усилий и денег.

6) могут быть разработаны различные уровни маркетинговых стратегий и может быть осуществлен целевой маркетинг, различные типы стратегий для различных потребностей клиента. Стратегия для покупателей в первый раз может быть настроена на увеличение их частоты, а также другая стратегия может быть четко настроена для постоянных клиентов, которые давно не покупали. Подобные действия могут иметь большое значение.

Всегда считается, что один лояльный клиент лучше, чем 100 иррациональных клиентов. Приведенный выше анализ явно помогает сделать более лояльных клиентов более эффективными. Усилия, которые были потрачены впустую раньше (подумайте) на привлечение 10 000 потенциальных клиентов, теперь используются для того, чтобы сосредоточиться и сконцентрироваться на 500 клиентах с высокими шансами, что является большим стимулом для доходов компании.

Анализ RFM может быть использован для улучшения маркетинга по электронной почте, запуска новых продуктов, настройки маркетинговых затрат, повышения вовлеченности пользователей и понимания клиентской базы компании и эффективного ее увеличения.

Почти все компании, занимающиеся электронной коммерцией, продуктами питания и блогами, используют этот тип анализа для увеличения доходов. Например, Flipkart или Amazon делают интенсивный маркетинг, основанный на стратегии, и предоставляют скидки/предложения, основанные на вышеизложенном. Zomato делает то же самое с самого начала, чтобы сохранить своего клиента, несмотря на растущую конкуренцию.

Несколько недостатков: многие клиенты покупают только один раз, усилия и признание их тратятся впустую, из-за интенсивного целевого маркетинга другие потенциальные крупные клиенты намеренно игнорируются, и не всегда статистика приводит к успеху.

Однако положительные стороны явно перевешивают отрицательные, и, следовательно, анализ RFM является лучшим способом по сравнению с другими примитивными методами.

**1.2 Обзор и анализ возможных альтернатив**

Сегментация рынка с использованием методов кластеризации очень распространена в:

– интернет-магазины, предлагающие товары, такие как ebay, amazon, avito.

– музыкальные приложения, предлагающие музыку как spotify,

– приложения для просмотра фильмов, такие как netflix, для составления предложений для фильмов, которые могут понравиться пользователю.

**1.3 Анализ функциональных и эксплуатационных требований**

**1.3.1 Стандарты**

Программный продукт разрабатывается на основании следующих государственных стандартов:

1. Межгосударственный стандарт ГОСТ 2018 «Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно – исследовательской работе. Структура и правила оформления».
2. Международный стандарт ISO/IEC 12207. Информационные технологии. Процессы жизненного цикла программного обеспечения.
3. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания.
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
5. ГОСТ 34.603-92. Информационная технология. Виды испытаний автоматизированных систем.

**1.3.2 Функциональные требования пользователя**

Программный продукт, разрабатываемый в рамках курсового проекта, должен удовлетворять следующему перечню функциональных требований:

– Формирование групп на основе кластеризации.

– Обеспечение доступа к истории покупок клиентов.

– Обеспечение просмотра различных кластеризованных групп с целью формирования предложений для рекламы.

**1.3.3 Входные данные**

Входными данными при работе с программным продуктом должны быть данные клиентской базы данных: имена, купленные продукти, даты покупкой. В формате xls. Входной язык приложения – английский.

**1.3.4 Выходные данные**

Выходные данные во время работы программы являются группы кластеризованных данных, отображаемые в формате xls.

**1.3.5 Требования к интерфейсу**

Главное меню программного продукта должно позволять пользователю выбрать данные о покупателе и выполнить кластеризацию этих данных.

Остальные окна программного продукта должны быть оснащены подсказками, а их функции быть интуитивно понятны пользователю.

Сообщения, выдаваемые программой при прохождении каких-либо действий, должны содержать краткое описание произведенных действий и содержать комментарии для облегчения дальнейшей работы пользователя.

**1.3.6 Требования к надежности**

При работе с программным продуктом необходимо предусмотреть:

* контроль вводимой информации, т.е. возможность отслеживания ошибок, допускаемых пользователем, и последующей реакции программы на них;
* предусмотреть блокировку некорректных действий пользователя при работе с системой.

**1.3.7 Требования к программной документации**

В состав сопровождающей документации программного продукта должны входить следующие компоненты:

* Пояснительная записка на 30 – 50 листах, содержащая описание разработки.
* В приложении к пояснительной записке исходные тексты основных модулей на языке Python
* Пояснительная записка, исходные тексты модулей
  + 1. **Требования к составу и параметрам технических средств**

Система должна работать на IBM совместимых персональных компьютерах. Минимал

– тип процессора –Intel®core™ i5-8300H;

* объем оперативного запоминающего устройства – 16 Мб;
* тип система – 64-bit Operating System
* тип монитора – HD (15').

**1.3.9 Модель вариантов использования**

**1.3.9.1 Диаграмма вариантов использования**

Действующие лица для диаграммы вариантов использования приведены

в таблице 1.

Таблица 1 **–** Действующие лица

|  |  |
| --- | --- |
| Термин | Значение |
| Сотрудник | Лицо, использующее программного продукта |

Таблица 2 –Таблица вариантов использования

|  |  |
| --- | --- |
| Сотрудник | Сотрудник продуктового магазина с необходимыми административными привилегиями, пользующийся услугами системы кластеризации |
| Ввод данных о покупатели | Ввод данных о покупатели в формате xls для кластеризации |
| Вход в систему | |  |  | | --- | --- | | Ввод пользователем логина и пароля для доступа к | | | системе (для Сотрудник) | |
| Вывод результатов | Вывода результатов в формате граф и таблицы |
| Выполнить кластеризацию | Выполняет кластеризацию и составляет результаты выполнение |
| Проверка корректности данных | Проверка корректности данных для того чтобы не произошли ошибки при кластеризации |

На основании всех выше рассмотренных вариантов использования была составлена диаграмма вариантов использования, представленная на рисунке 3.

**1.3.9.2 Описание варианта использования «Выполнить кластеризацию»**

Действующие лица: сотрудник.

Заинтересованные лица и их требования:

– сотрудник хочет выполнить кластеризации над данными о покупателях;

Предусловия: клиент должен войти в систему

Постусловия: если вариант использования выполнен успешно, пользователь выполняет кластеризации, сохраняет результаты. В противном случае состояние системы не изменяется.

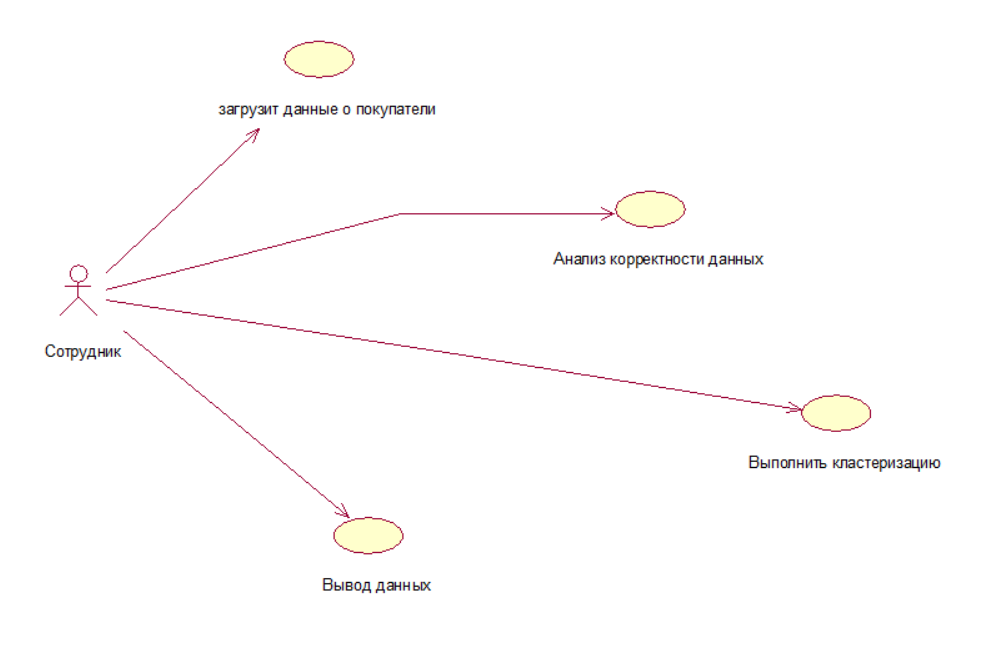


Рисунок 3 – Диаграмма вариантов использования «Выполнить кластеризации»

Основной сценарий.

1. Сотрудник выбирает «Clusterize».

2.Система выполняет кластеризацию над данными.

3. Система отображает отчеты в виде графов и таблиц.

4. Система предлагает сохранить результаты.

5. Сотрудник выбирает сохранение результатов.

5а. Если сотрудник выбирает не сохранять результаты, то вариант использования завершается.

**2 Проектирование информационной системы**

**2.1 Разработка архитектуры системы**

Разрабатываемое приложение представляет собой локальное приложение.

Персональные компьютеры должны быть расположены, чтобы сотрудник мог работать с приложением и принтером.

На рисунке 4 представлена предварительная схема развертывания разработанного приложения - архитектура аппаратного обеспечения системы.

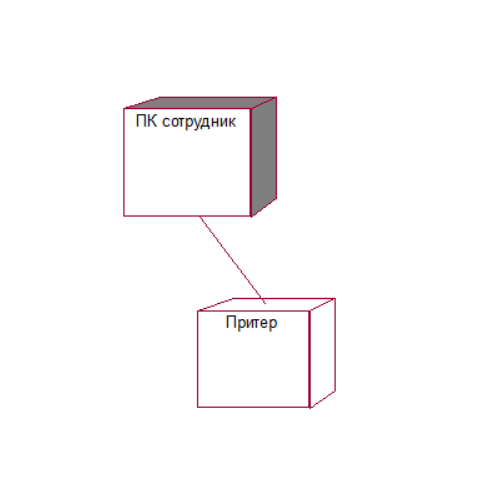


Рисунок 4 – Архитектура технических средств системы

**2.2 Разработка модели предметной области**

Пользуясь списком категорий и методом анализа словесного описания вариантов использования, составлен список на роль концептуальных классов для предметной области. Он соответствует требованиям и принятым упрощенным системам для всей предметной области.

Список концептуальных классов:

* Xls документ;
* Анализ;
* Результат работы;
* Результирующий документ;
* Результирующий граф;
* Кластер.
* Сотрудник.
* Покупка

На основании анализа словесного описания варианта использования, составлен список ассоциаций для предметной области, представленный в таблице 4.

Таблица 4 – Ассоциации для модели предметной области

|  |  |
| --- | --- |
| Ассоциация | Описание ассоциации |
| Проводится по | Анализ проводится по покупкам |
| Вычисляется | По результирующему xls документ вычисляются кластеры |
| Выполняет | Сотрудник выполняет анализ |
| Является результатом | Результирующий xls документ является результатом анализа |
| включает | Кластеры включает результирующий график |
| включает | Кластеры включает результирующий таблицу |
| включает | результирующий таблицу включает результаты работы |
| включает | результирующий график включает результаты работы |

На основании анализа технического задания и описания вариантов использования выделены атрибуты классов для модели предметной области, представленные в таблице 5.

Таблица 5 – Атрибуты классов для модели предметной области

|  |  |
| --- | --- |
| Название класса | Атрибуты класса |
| Анализ | Столбец даты покупки  Столбец с именами клиентов  Столбец выручки клиента |
| Покупка | Название компания  Количество покупок  Дата покупки  Тип покупки |
| Сотрудник | ФИО  Дата рождение  Название компания  Позиция в компании |
| Кластер | координаты точек |
| Результирующий таблицы | столбцы таблицы |
| Результирующий график | График (Graph.png) |
| Результирующий xls документ | xls документ после Анализа |
| Резултаты работы | 3D рисунки в формате png, данные сотрудник выполняющий кластееризации и название магазина, дата. |

В результате объединения концептуальных классов, ассоциаций и атрибутов классов концептуальная модель предметной области имеет вид, показанный на рисунке 5

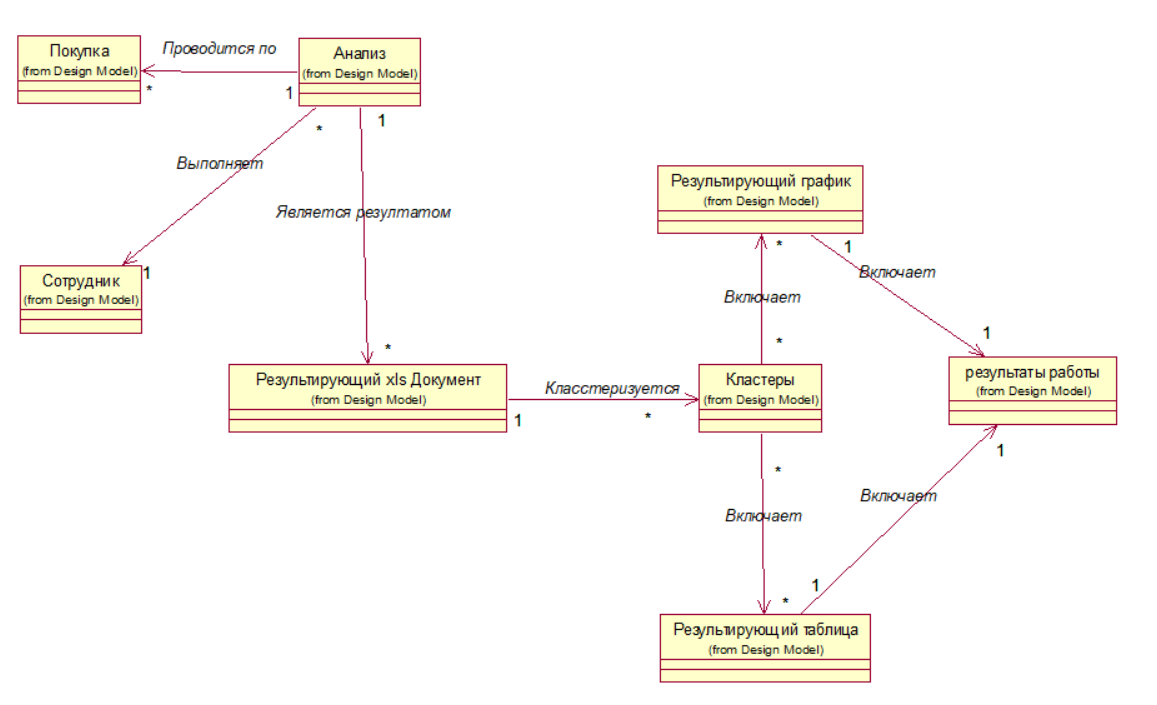


Рисунок 5 – Концептуальная модель предметной области

**2.3 Разработка алгоритма функционирования системы**

Вход в систему осуществляется сотрудником без проверки имении пароля.

Алгоритм работы системы в виде диаграммы деятельности приведен на рисунке 6.

В главном меню пользователю дается на выбор несколько действий: «Загрузить», «Анализировать», «Кластеризовать данные», «Сохранить» и «Выход».

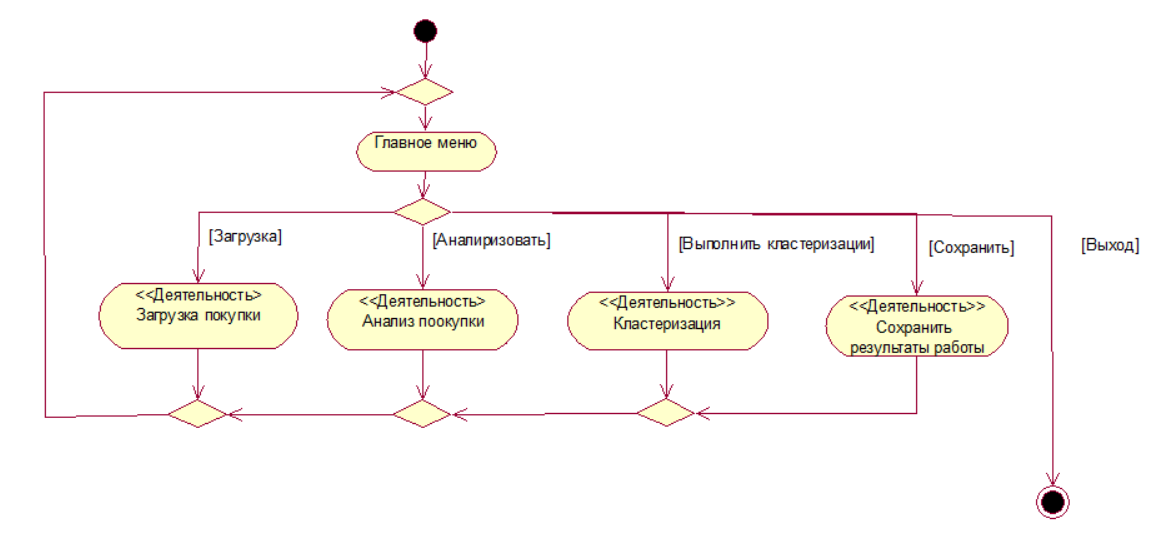


Рисунок 6 – Диаграмма деятельностей «Деятельность алгоритм работы системы»



Рисунок 7 – Диаграмма деятельностей «Деятельность кластеризация»

**2.4 Проектирование интерфейса пользователя**

На основании алгоритма функционирования и требований к интерфейсу (см. раздел 1) разработана диаграмма состояний (рис. 7).

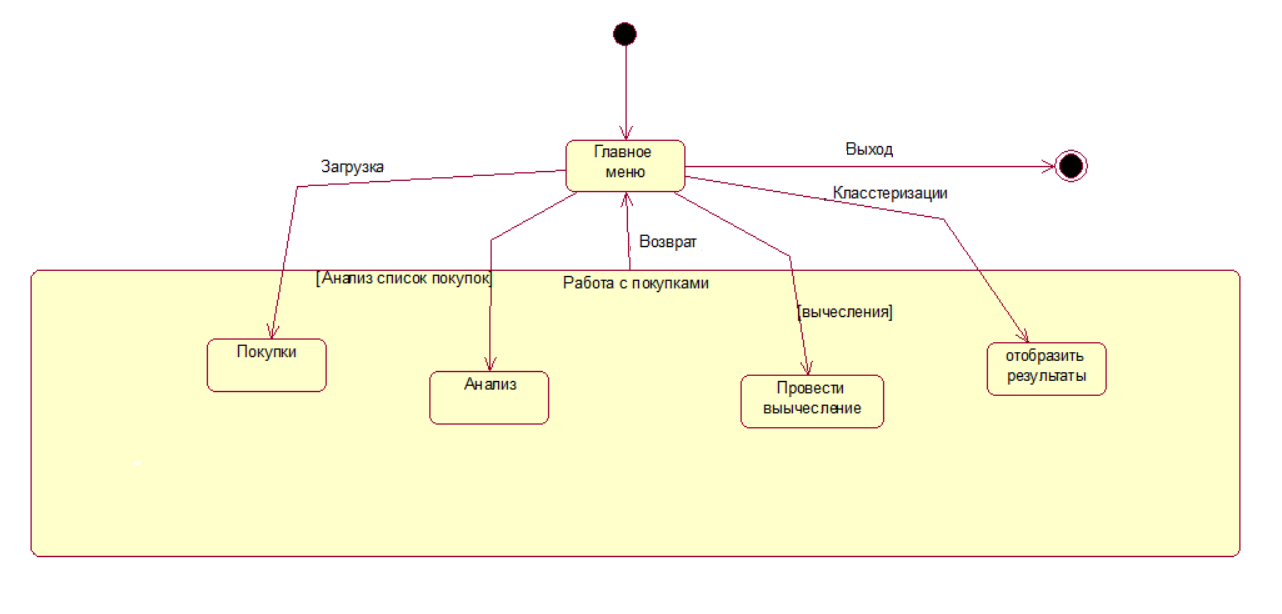


Рисунок 8 – Диаграмма состояний интерфейса

**2.5 Реляционная модель данных**

На рисунке 8 изображена реляционная модель данных.

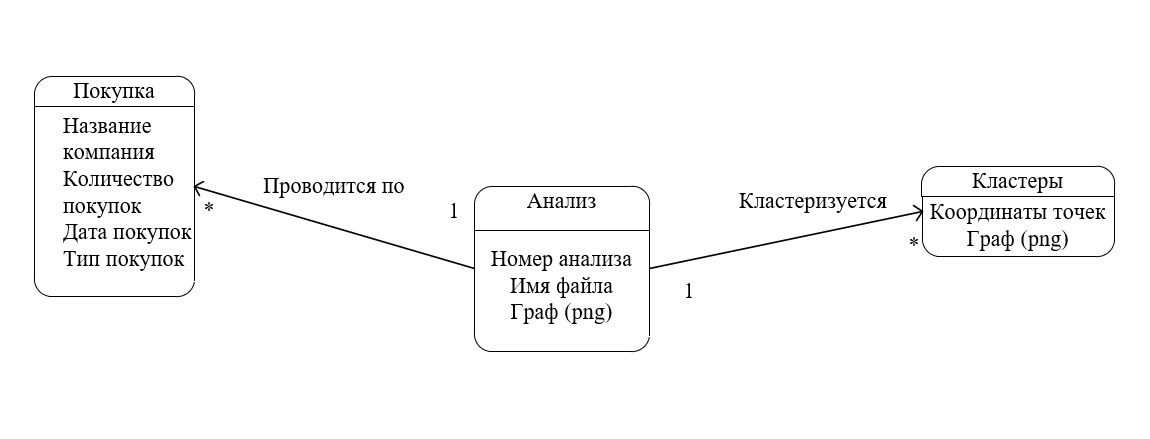


Рисунок 9 – Реляционная модель данных

Реляционная модель данных разработана на основе концептуальной модели предметной области. Реляционная модель данных в дальнейшем служит для разработки БД.

## **2.6. Проектирование классов предметной области**

## **2.6.1. Построение диаграмм последовательностей для варианта использования «Выполнить кластеризацию»**

Диаграмма последовательности, описывающая основной поток событий изображена на рисунке 9.

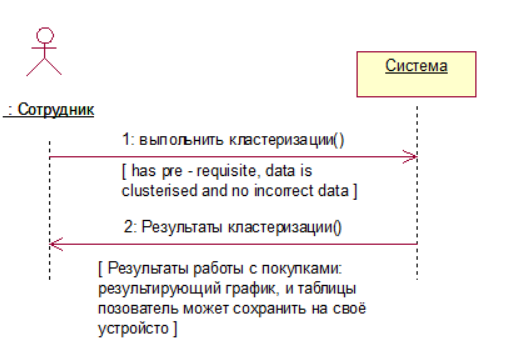


Рисунок 10 – Диаграмма последовательности системных операций варианта использования «выполнить кластеризации»

## **2.6.2 Построение диаграммы кооперации**

Структурные особенности передачи и приема сообщений между объектами представлены на диаграмме кооперации на рисунке 10.

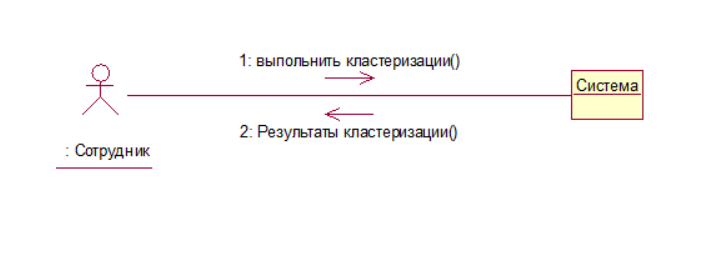


Рисунок 11 – Диаграмма кооперации

## **3 Реализация системы**

## **3.1 Реализация программного обеспечения системы**

## **3.1.1 Разработка диаграммы компонентов**

Реализация программного обеспечения системы представлена на рисунке 11 в виде Главная форма. Она определяет архитектуру разрабатываемой системы на физическом уровне и представляет зависимости между программными компонентами.

## **3.1.2 Объекты интерфейса пользователя**

Программный продукт состоит из одной формы. Внешний вид формы программы (Main) представлен на рисунке 11.

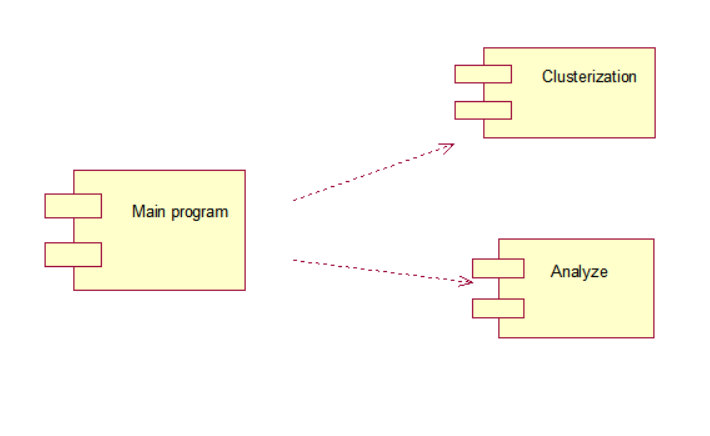


Рисунок 11 – Диаграмма компонентов приложения



Рисунок 12 – Главная форма

В таблице 9 представлены расположенные на форме MAIN компоненты.

Таблица 18 – Компоненты формы MAIN

|  |  |  |  |
| --- | --- | --- | --- |
| №№ | Наименование компонента | Тип компонента | Назначение |
| 10 | menuStrip1 | MenuStrip | Панель инструментов |
| 21 | Tab1 | TableWidget1 | Логин и регистрация |
| 32 | Tab2 | TableWidget1 | Главная страница |
| 33 | Tab3 | TableWidget1 | Для отображение загруженных данных |
| 44 | Tlabel1 | Tlabel | Текущий открытый файл |
| 65 | Tlabel2  Tlabel3  Tlabel4 | Tlabel | Недавно открытые файли |
| 66 | TButton1 | TButton | Загрузить файл |
| 77 | TButton2 | TButton | Сохранить файл |
| 88 | Combobox1 | TCombobox | Выбрать столбец id покупателей |
| 99 | Combobox2 | TCombobox | Выбрать столбец дата покупок |
| 110 | Combobox3 | TCombobox | Выбрать столбец доходов |
| 111 | TButton3 | TButton | Выполнить анализ |
| 112 | Combobox4 | TCombobox | Выбрать число recency |
| 113 | Combobox5 | TCombobox | Выбрать число frequency |
| 114 | Combobox6 | TCombobox | Выбрать число Monetary |
| 115 | TButton4 | TButton | Выполнить кластеризации |

### 

## **3.2 Реализация технического обеспечения**

## **4 Анализ результатов**

## **4.1 Разработка тестов и тестирование системы**

Создание теста для метода RFM\_dataframe класса Analyze() модули Analysis используя библиотеки unitest и pandas.testing в питоне.

Чтобы использовать unittest в тестовом классе, он должен быть унаследован от того класса, в котором мы собираемся его использовать, например:

*class TestApp(unittest.TestCase):*

Имена методов в тестовом классе должны предшествовать слова Test\_, чтобы класс распознал, что метод является тестовой функцией.

В первом тестовом примере функция Analyze возвращает фрейм данных RFM - Recency, Frequency и Monetary dataframe. Было бы обременительно тестировать фрейм данных с помощью unitest, поэтому используется pandas.testing. Pandas.testing имеет специальные функции, которые позволяют сравнивать свойства двух фреймов данных, просто изменяя параметры функций.

Входной данных:

Состоит из столбцов идентификации; данные о покупателях, даты покупки, сумма, потраченная конкретным покупателем на конкретную дату.

*data =*

*{'Order ID': ['CA-2016-152156', 'CA-2016-152156','CA-2016-138688','US-2015-108966','US-2015-108966','CA-2014-115812','CA-2014-115812'],*

*'Order Date': ['11/8/2016', '11/8/2016', '11/8/2016','10/11/2015','10/11/2015','6/9/2014','6/9/2014'],*

*'Profit': ['261.96', '731.94','14.62','957.5775','22.368','48.86','7.28']*

*}*

Ожидаемые выходные данные:

Состоит из; строка идентификации клиента, частота посещений клиентов, время последнего посещения клиента и доход, который принес клиент.

*result =*

*{'Order ID': ['CA-2016-152156','CA-2016-138688','US-2015-108966','CA-2014-115812'],*

*'Frequency': [1, 0, 1, 1],*

*'Recency': [0, 0,394, 883],*

*'logRevenue': [496.95000, 14.62000, 489.97275, 28.07000]*

*}*

Класс Analyze имеет один основной метод, который возвращает фрейм данных, 4 теста выполняются на результирующем фрейме данных. Для тестирования результирующего фрейма данных используется метод assert\_dataframe\_equal из panadas.testing, этот метод имеет много параметров, но из них только четыре являются полезными в данном случае.

Тест 1

Эта функция предназначена для сравнения двух DataFrames и вывода любых различий.

*def test\_Analyze\_1(self): pd\_testing.assert\_frame\_equal(Analyze(data\_df,1,2,3).RFM\_dataframe(),result\_df,False,False,False,False)*

Тест 2

Проверять, идентичен ли dtype DataFrame.

*def test\_Analyze\_2(self):*

*pd\_testing.assert\_frame\_equal(Analyze(data\_df,1,2,3).RFM\_dataframe(),result\_df,True,False,False,False)*

Тест 3

Следует ли проверять класс столбцов, dtype и inferred\_type идентичны.

*def test\_Analyze\_3(self):*

*pd\_testing.assert\_frame\_equal(Analyze(data\_df,1,2,3).RFM\_dataframe(),result\_df,False,False,True,False)*

Тест 4

Проверять, идентичен ли класс DataFrame.

*def test\_Analyze\_4(self): pd\_testing.assert\_frame\_equal(Analyze(data\_df,1,2,3).RFM\_dataframe(),result\_df,False,False,False,True)*

После завершения тестирования открывается окно обозревателя тестов, в котором отображаются результаты (рисунок 13).

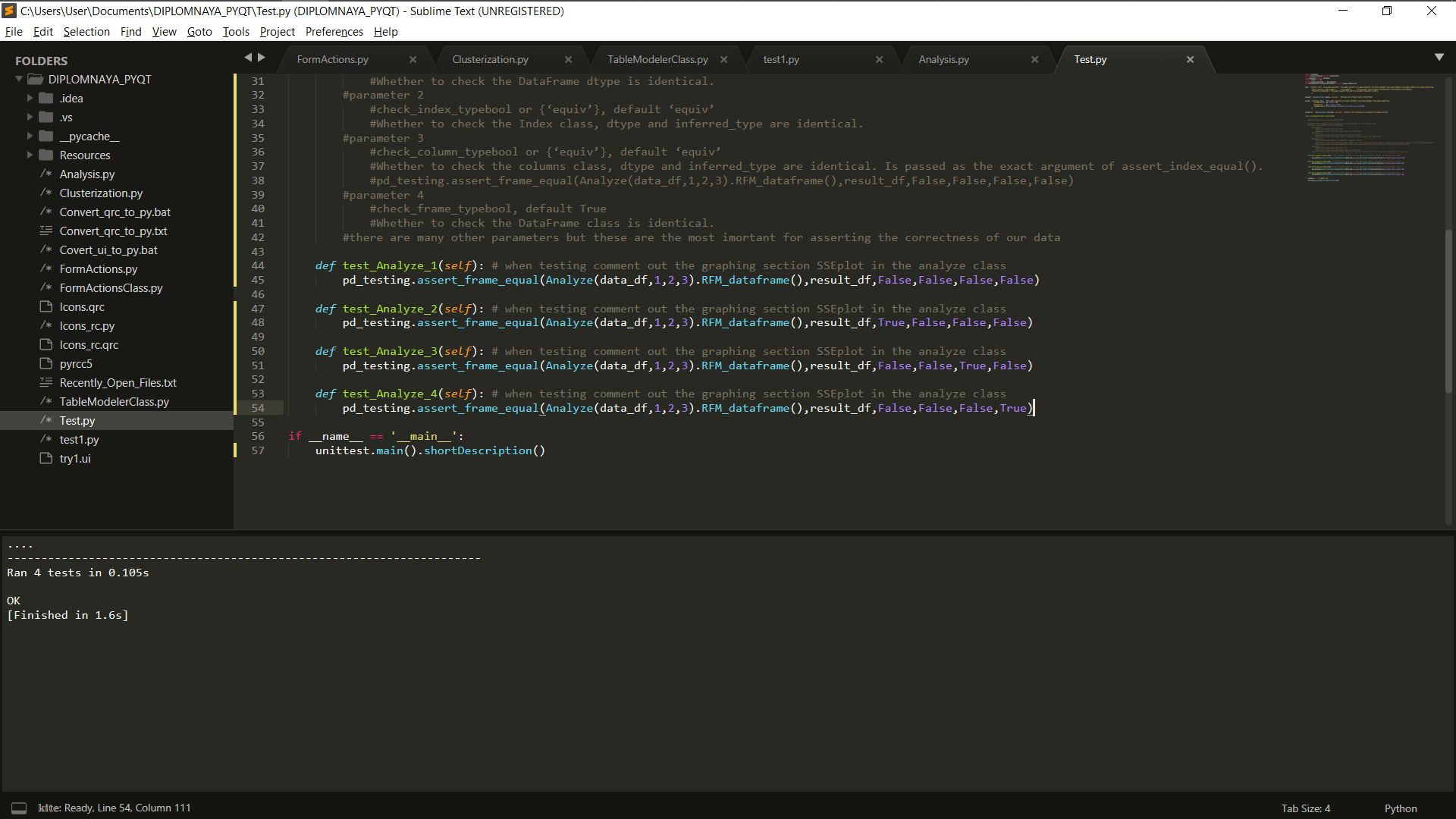


Рисунок 13 – Вывод успешно выполненного теста

## **4.2 Анализ эргономических эффектов от использования системы**

## **ЗАКЛЮЧЕНИЕ**

Разработанный программный продукт позволяет автоматизировать процесс системы сегментировать покупатели продуктового магазина.

В процессе создания системы в соответствии с заданием были разработаны: модель вариантов использования, концептуальная модель предметной области, диаграммы деятельности, реляционная модель данных, диаграмма состояний интерфейса, формы интерфейса и диаграмма компонентов.

В данном курсовом проекте было проведено программирование на языке Python. Главное меню, события кнопок, визуальная составляющая проекта прописаны в qt Designer, функции выполняющий кластеризации были прописаны на языке Python 3.9.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Обзор алгоритмов кластеризации [Cайт]. URL: https://scikit-learn.org/stable/modules/clustering.html c (дата обращения: 01.10.2020).
2. Интерфейс на питон [Cайт]. URL: / Обзор алгоритмов кластеризации [Cайт]. URL: https://www.riverbankcomputing.com/static/Docs/PyQt5/ (дата обращения: 01.10.2020).
3. Обзор алгоритмов кластеризации [Cайт]. URL: https://scikit-learn.org/stable/modules/clustering.html (дата обращения: 01.03.2021).
4. RFM алгоритмов [Cайт]. URL: <https://www.ibm.com/docs/en/spss-statistics/27.0.0?topic=analysis-rfm-scores-from-customer-data> (дата обращения: 01.01.2021).
5. PyQt табличный структуры [Cайт]. URL: <https://www.mfitzp.com/tutorials/qtableview-modelviews-numpy-pandas/> (дата обращения: 16.05.2021).
6. Использование метода локтя для определения оптимального количества кластеров для кластеризации k-средних [Cайт]. URL: <https://bl.ocks.org/rpgove/0060ff3b656618e9136b> (дата обращения: 01.11.2020).
7. Наборы данных[Cайт]. URL: <https://www.kaggle.com/vijayuv/onlineretail> (дата обращения: 01.02.2021).
8. RFM-анализ для сегментации клиентов и маркетинга лояльности [Cайт]. URL: <https://www.youtube.com/watch?v=OYohJxp2l9k> (дата обращения: 25.01.2021).

**ПРИЛОЖЕНИЕ А**

**Текст программы**

# import required libraries for dataframe and visualization

import numpy as np

import pandas as pd

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.pyplot as plt

import datetime as dt

# import required libraries for clustering

import sklearn

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score

from scipy.cluster.hierarchy import linkage

from scipy.cluster.hierarchy import dendrogram

from scipy.cluster.hierarchy import cut\_tree

"""

Recency refers to the interval between the time, that the latest consuming behavior happens,

and present. Many direct marketers believe that most-recent purchasers are more likely to

purchase again than less-recent purchasers. Frequency is the number of transactions that a

customer has made within a certain period. This measure is used based on the assumption

that customers with more purchases are more likely to buy products than customers with

fewer purchases. Monetary refers to the cumulative total of money spent by a particular

customer.

"""

class Clusterize:

# Reading the data on which analysis needs to be done

def \_\_init\_\_(self,freq\_k, recen\_k, rev\_k, rfm\_df ):

self.rfm\_df = rfm\_df

self.rfm\_df = self.frequency\_cluster(freq\_k, 'Frequency', self.rfm\_df)

self.rfm\_df.groupby('FrequencyCluster')['Frequency']

self.rfm\_df = self.recency\_cluster(recen\_k, 'Recency', self.rfm\_df)

self.rfm\_df.groupby('RecencyCluster')['Recency']

self.rfm\_df = self.revenue\_cluster(rev\_k, 'logRevenue', self.rfm\_df)

self.rfm\_df.groupby('RevenueCluster')['logRevenue']

self.rfm\_df['OverallScore'] = self.rfm\_df['RecencyCluster'] + self.rfm\_df['FrequencyCluster'] + self.rfm\_df['RevenueCluster']

self.rfm\_df.groupby('OverallScore')['Recency','Frequency','logRevenue'].mean()

self.graphClusters()

#self.graphIndividaulMeasures()

def order\_cluster(self,cluster\_col, feature\_col, df, ascending):

df\_new = df.groupby(cluster\_col)[feature\_col].mean().reset\_index()

df\_new = df\_new.sort\_values(by=feature\_col,ascending=ascending).reset\_index(drop=True)

df\_new['index'] = df\_new.index

df\_final = pd.merge(df,df\_new[[cluster\_col,'index']], on=cluster\_col)

df\_final = df\_final.drop([cluster\_col],axis=1)

df\_final = df\_final.rename(columns={"index":cluster\_col})

return df\_final

def frequency\_cluster(self, cluster\_number, frequency\_col, dataframe):

frequency\_kmeans = KMeans(n\_clusters=cluster\_number)

frequency\_kmeans.fit(dataframe[[frequency\_col]])

# Assigning cluster prediction to customers

dataframe['FrequencyCluster'] = frequency\_kmeans.predict(dataframe[[frequency\_col]])

# Ordering clusters from low to high and identifying statistics

dataframe = self.order\_cluster('FrequencyCluster', frequency\_col, dataframe, True)

return dataframe

def recency\_cluster(self, cluster\_number, recency\_col, dataframe):

recency\_kmeans = KMeans(n\_clusters=cluster\_number)

recency\_kmeans.fit(dataframe[[recency\_col]])

# Assigning cluster prediction to customers

dataframe['RecencyCluster'] = recency\_kmeans.predict(dataframe[[recency\_col]])

# Ordering clusters from low to high and identifying statistics

dataframe = self.order\_cluster('RecencyCluster', recency\_col, dataframe, False)

return dataframe

def revenue\_cluster(self, cluster\_number, revenue\_col, dataframe):

revenue\_kmeans = KMeans(n\_clusters=cluster\_number)

revenue\_kmeans.fit(dataframe[[revenue\_col]])

# Assigning cluster prediction to customers

dataframe['RevenueCluster'] = revenue\_kmeans.predict(dataframe[[revenue\_col]])

# Ordering clusters from low to high and identifying statistics

dataframe = self.order\_cluster('RevenueCluster', revenue\_col,dataframe,True)

return dataframe

def graphClusters(self):

# Naming and defining segments

self.rfm\_df['Segment'] = 0

self.rfm\_df.loc[self.rfm\_df['OverallScore']>4,'Segment'] = 1

self.rfm\_df.loc[self.rfm\_df['OverallScore']>6,'Segment'] = 2

high = self.rfm\_df.query('Segment == 2')

mid = self.rfm\_df.query('Segment == 1')

low = self.rfm\_df.query('Segment == 0')

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

g1= (low['Frequency'].values, low['Recency'].values, low['logRevenue'].values)

g2 = (mid['Frequency'].values, mid['Recency'].values, mid['logRevenue'].values)

g3= (high['Frequency'].values, high['Recency'].values, high['logRevenue'].values)

data = [g1, g2, g3]

colors = ['#440154FF', '#20A387FF', '#FDE725FF']

groups = ['Low', 'Med', 'High']

for data, color, group in zip(data, colors, groups):

x, y, z = data

ax.scatter(x, y, z, alpha=0.5, c=color, label=group)

# Make legend

ax.legend()

ax.set\_xlabel('Frequency')

ax.set\_ylabel('Recency')

ax.set\_zlabel('Revenue')

ax.set\_title('Spatial Representation of Segments', loc='left')

plt.show();

def graphIndividaulMeasures(self):

plt.figure()

plt.scatter(list(self.rfm\_df['RecencyCluster']))

plt.show();