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.

Допустить к защите:
и.о. заведующего кафедрой
_____ /Макаров К.С./
(подпись)

«____» _____ 20____ г.

Курск, 2021

**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

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

Кластеризация – это задача разделения совокупности данных на несколько групп таким образом, чтобы совокупности данных в группах были более похожи на другие совокупности данных в той же группе и отличались от совокупности данных в других группах. Кластеризация очень важна, поскольку она определяет внутреннюю группировку среди имеющихся немаркированных данных. Из-за вышеупомянутой важности можно использовать в интернет-магазине продуктов для группировки клиентов в разные группы в зависимости от различных факторов [1]:

– 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 выделенных 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-то конкретное время года.

Целью разработки приложения является создание системы кластеризации, которая группирует клиентов интернет-магазина 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 выделенных на основе кластеризации групп с целью формирования предложений для рекламы.

Программный продукт предназначен для использования в любом онлайн магазине.

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.

Аннотация: в выпускной квалификационной работе рассмотрены основные моменты проектирования и разработки системы кластеризации пользователей магазина. группирует клиентов интернет-магазина продуктов в несколько групп и дает предложения о том, какие продукты рекламировать для каждой результирующей группы.

Abstract: in the final qualifying work, the main points of design and development of a clustering system for store users are considered. groups customers of an online product store into several groups and gives suggestions on which products to advertise for each resultant group.

# 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 удобным и снижает шансы клиента или пользователя сайта купить столько продуктов, сколько они купили бы, если бы им были предоставлена реклама необходимых продуктов.

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

Система позволит обеспечить:

– предоставление лучших индивидуальных предложений по продукту для каждого отдельного пользователя;

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

– предоставление важных данных о пользователях менеджеру продуктового магазина, что поможет ему определить, какими и когда продуктами следует 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 и рисунке В.10 представлена контекстная DFD-диаграммы системы интеллектуальная система кластеризации для продуктового магазина в BPWin. Входными данными служит информация о покупателях (Market segmentation system) с выходными графиками (Graph) и таблицами (Table) с результатами анализа.

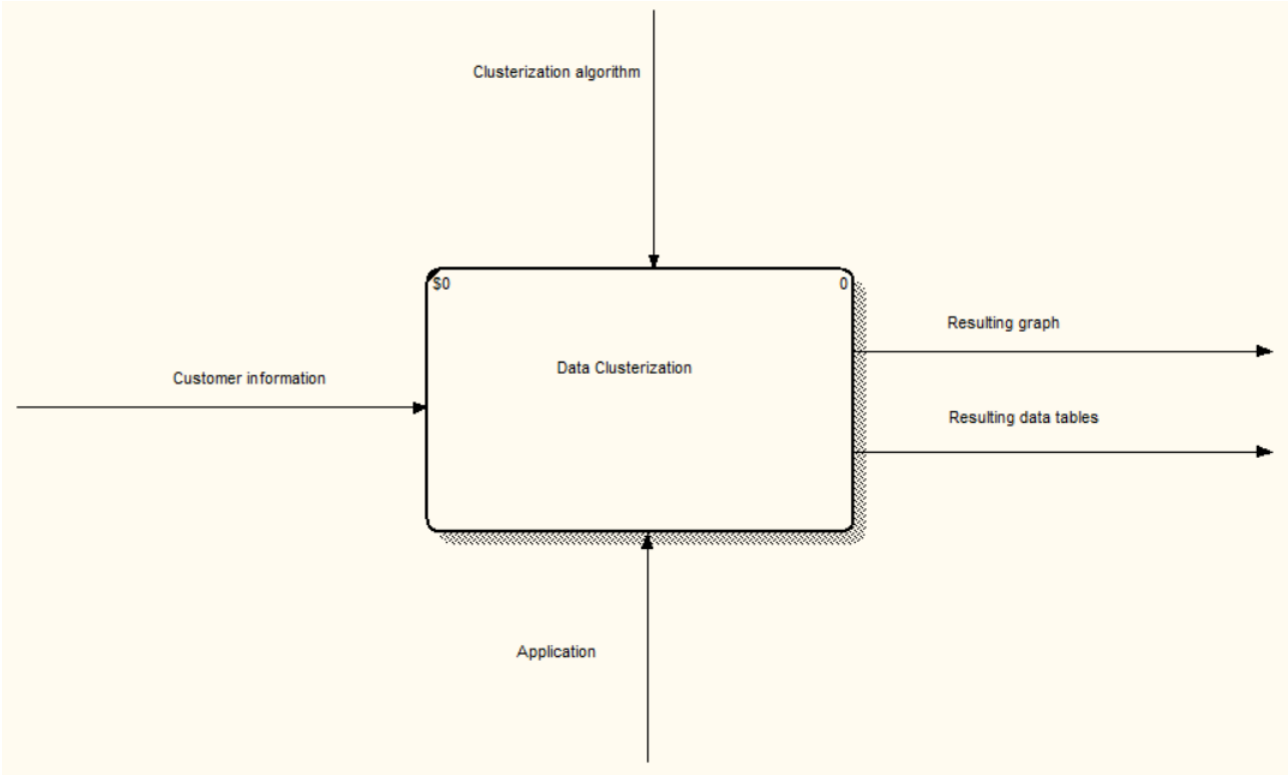



Рисунок 1 – Контекстная SADT-диаграмма системы

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

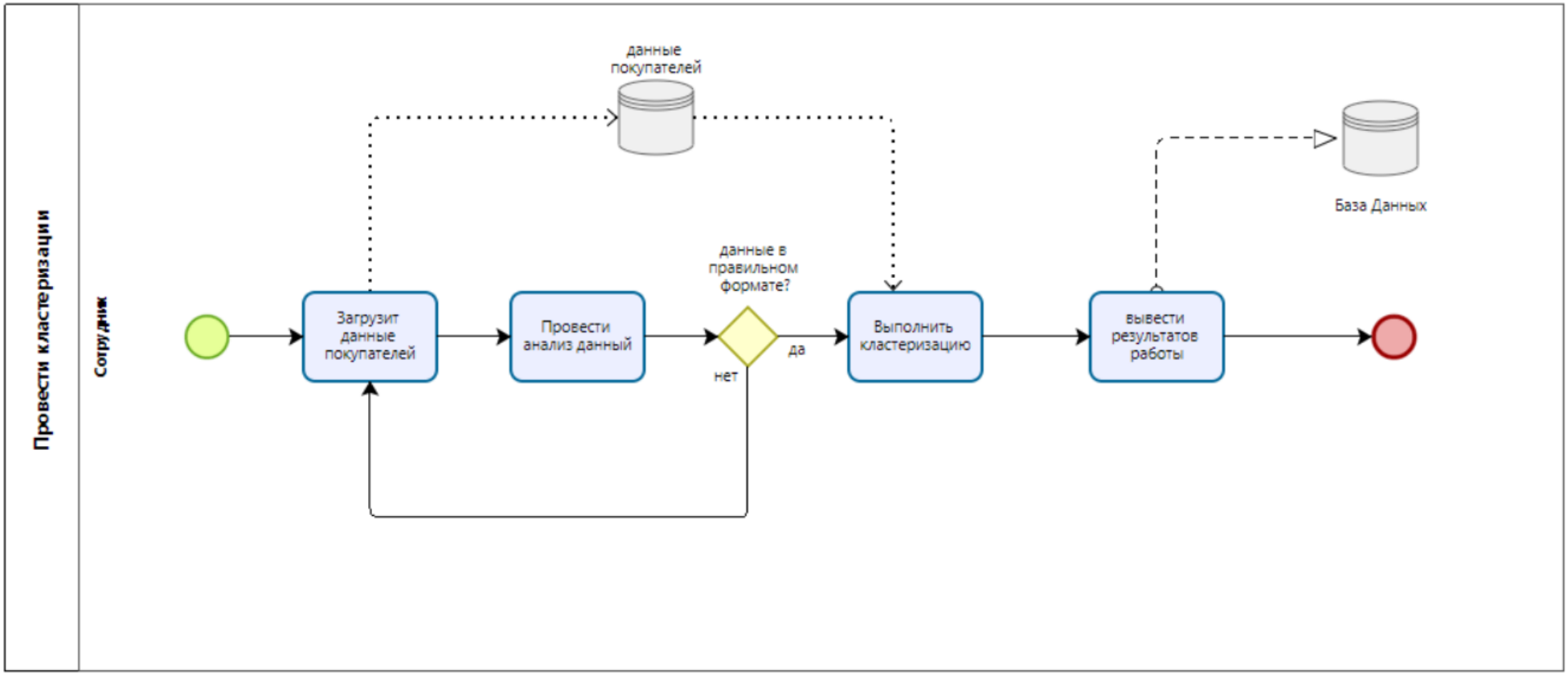



Рисунок 2 – Описание бизнес-процесса «Заключение договора» в BPMN

### 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. 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 [2].

Алгоритм K-means сводится к выполнению следующих шагов:

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 Алгоритм анализа RFM

Алгоритма K-means кластеризации основан на анализ RFM. Суть анализа RFM (Давность, Частота, Денежная стоимость) состоит в том, чтобы разделить клиентов на группы на основе того, как давно они совершили свою последнюю покупку, как часто они покупают товары и какова средняя стоимость их заказов. Для каждой из этих метрик мы назначаем клиентов в одну из трех групп, которым присваивается номер от 1 до 3.

Давность:

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 выгоден для любой компании, 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, во-вторых, анализ RFM в сочетании с различными другими показателями приоритета может иметь первостепенное значение для поиска моделей поведения клиентов, что также важно для маркетинга, целей продаж и т. д.

Можно выбрать различные способы маркетинга, и затраты могут быть сокращены за счет огромной маржи, поскольку компании не будут тратить

время на дешевых клиентов по сравнению с почти потерянными/потерянными клиентами или начинающими клиентами [3 - 5].

Благодаря анализу RFM денежные лояльные клиенты могут способствовать тому, чтобы компания работала с высококачественным, немного более дорогим продуктом вместо того, чтобы снижать стоимость, приводящую к снижению качества продукции

Вознаграждение лояльных или потенциально лояльных клиентов может быть осуществлено путем предоставления им бесплатной доставки или предоставления небольших бесплатных услуг или ваучеров, 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 сохраняются обе категории, не тратя много усилий и денег.

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 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 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 [6 - 8].

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.[9]

## 1.2 Обзор и анализ возможных альтернатив

Интеллектуальная система кластеризации для продуктового магазина очень распространена в таких торговых структурах, как:

– 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 продукты, даты покупок. 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;

– 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:

1. Пояснительная записка на 40 – 60 листах, содержащая описание разработки.

2. В приложении к пояснительной записке исходные тексты основных модулей на языке Python.

3. Пояснительная записка, исходные тексты модулей.

### 1.3.8 Требования к составу и параметрам технических средств

Система должна работать на IBM совместимых персональных компьютерах. Минимальные требования:

– 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.

Варианты использования для диаграммы вариантов использования приведены в таблице 2.

Таблица 1 – Действующие лица

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

Таблица 2 –Таблица вариантов использования

| Термин | Значение |
|---|---|
| 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 и рисунке В.2.

### 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.

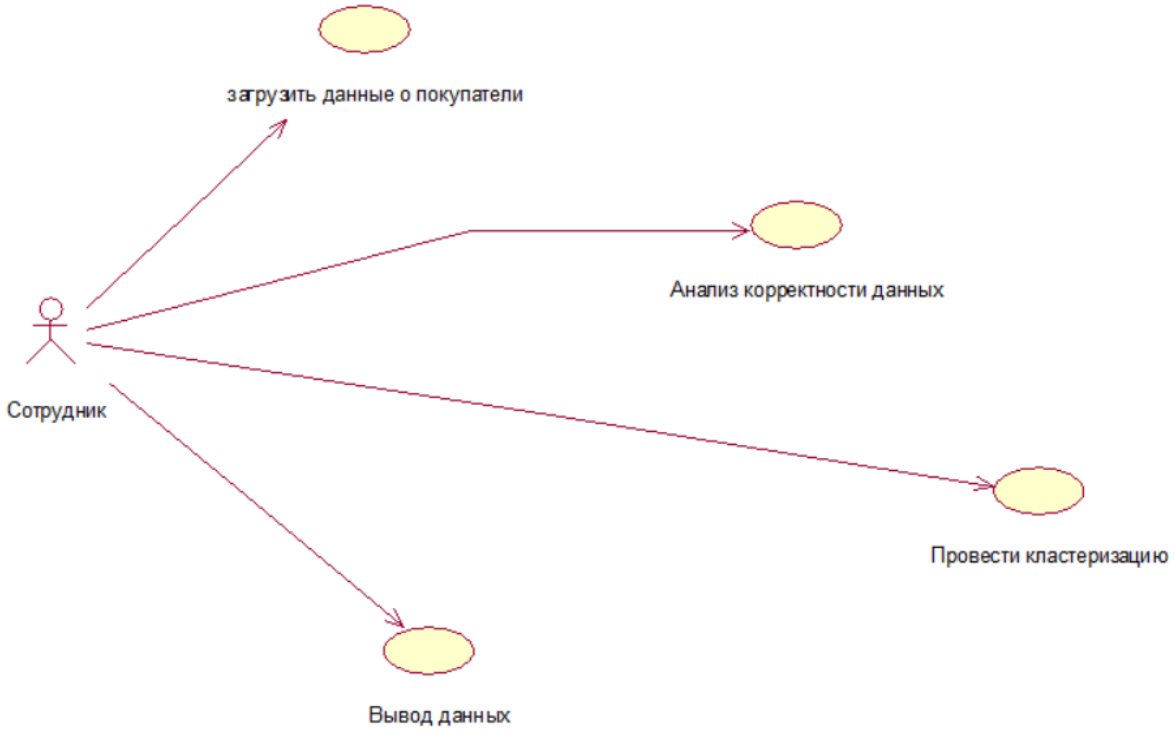



Рисунок 3 – Диаграмма вариантов использования «Выполнить кластеризации»

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.

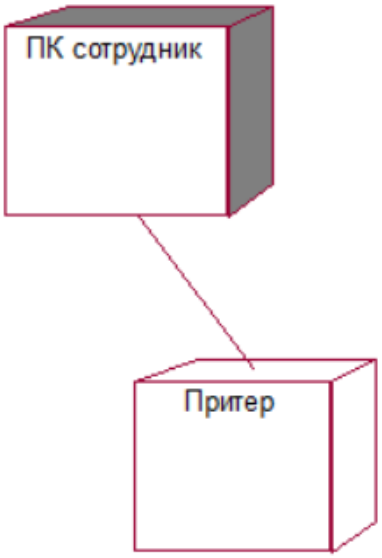



Рисунок 4 – Архитектура технических средств системы

## 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 3.

Таблица 3 – 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 |

Продолжение таблицы 3

| Ассоциация | Описание ассоциации |
|---|---|
| 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 4.

Таблица 4 – Атрибуты классов для модели предметной области

| Название класса | Атрибуты класса |
|---|---|
| Анализ | Столбец даты покупки<br>Столбец с именами клиентов<br>Столбец выручки клиента |
| Покупка | Название компания<br>Количество покупок<br>Дата покупки<br>Тип покупки |
| Сотрудник | ФИО<br>Дата рождение<br>Название компания<br>Позиция в компании |
| Кластер | координаты точек |
| Результирующий таблицы | столбцы таблицы |
| Результирующий график | График (Graph.png) |

Продолжение таблица 4

| Название класса | Атрибуты класса |
|---|---|
| Результирующий xls документ | xls документ после Анализа |
| Результаты работы | 3D рисунки в формате png, данные сотрудник, выполняющий кластеризации и название магазина, дата. |

В результате объединения концептуальных классов, ассоциаций и атрибутов классов концептуальная модель предметной области имеет вид, показанный на рисунке 5 и рисунок В.4.

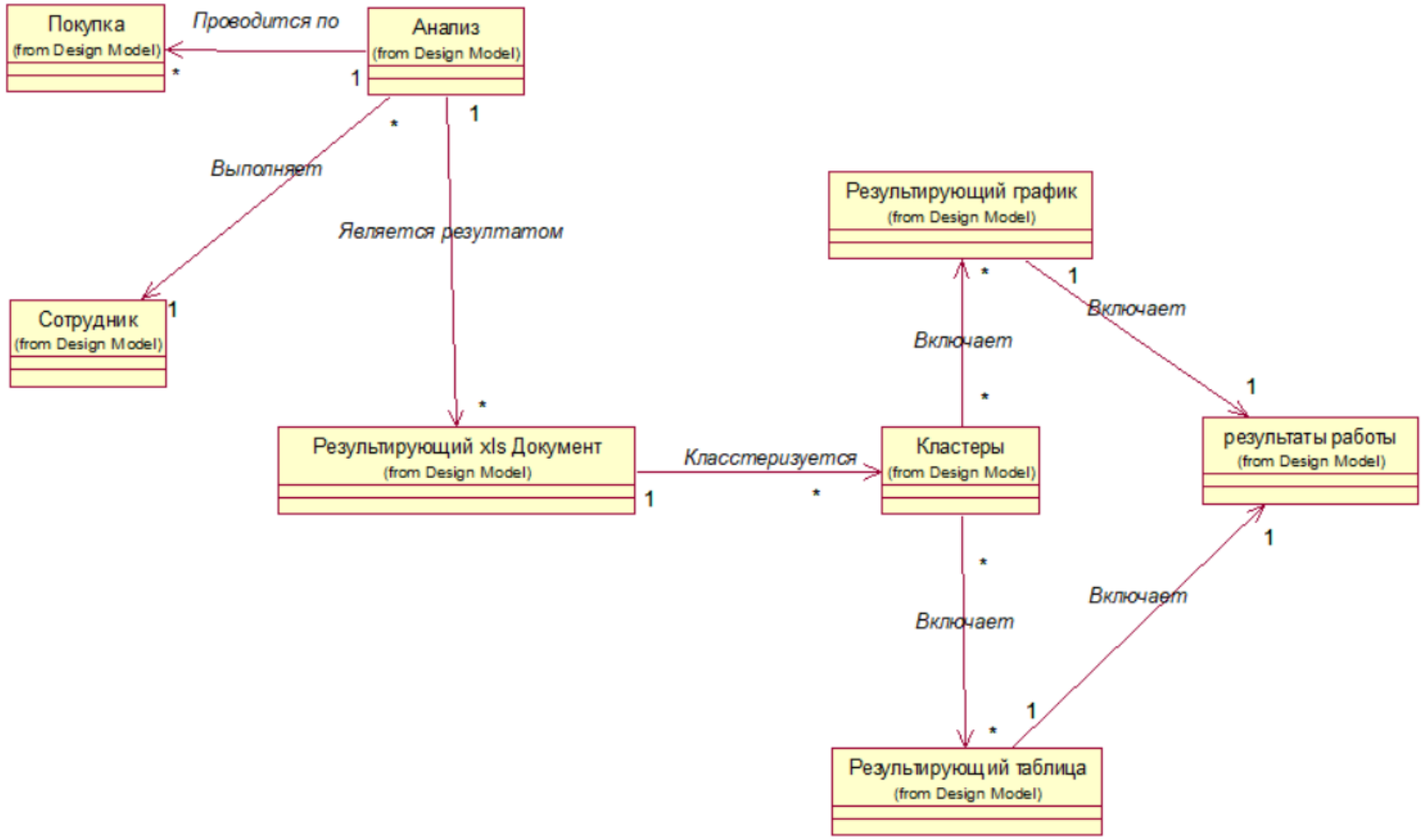



Рисунок 5 – Концептуальная модель предметной области

## 2.3 Разработка алгоритма функционирования системы

Вход в систему осуществляется сотрудником без проверки имени и 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».

Диаграмма деятельностей «Деятельность кластеризация» приведен на рисунке 7 и рисунке В.3.

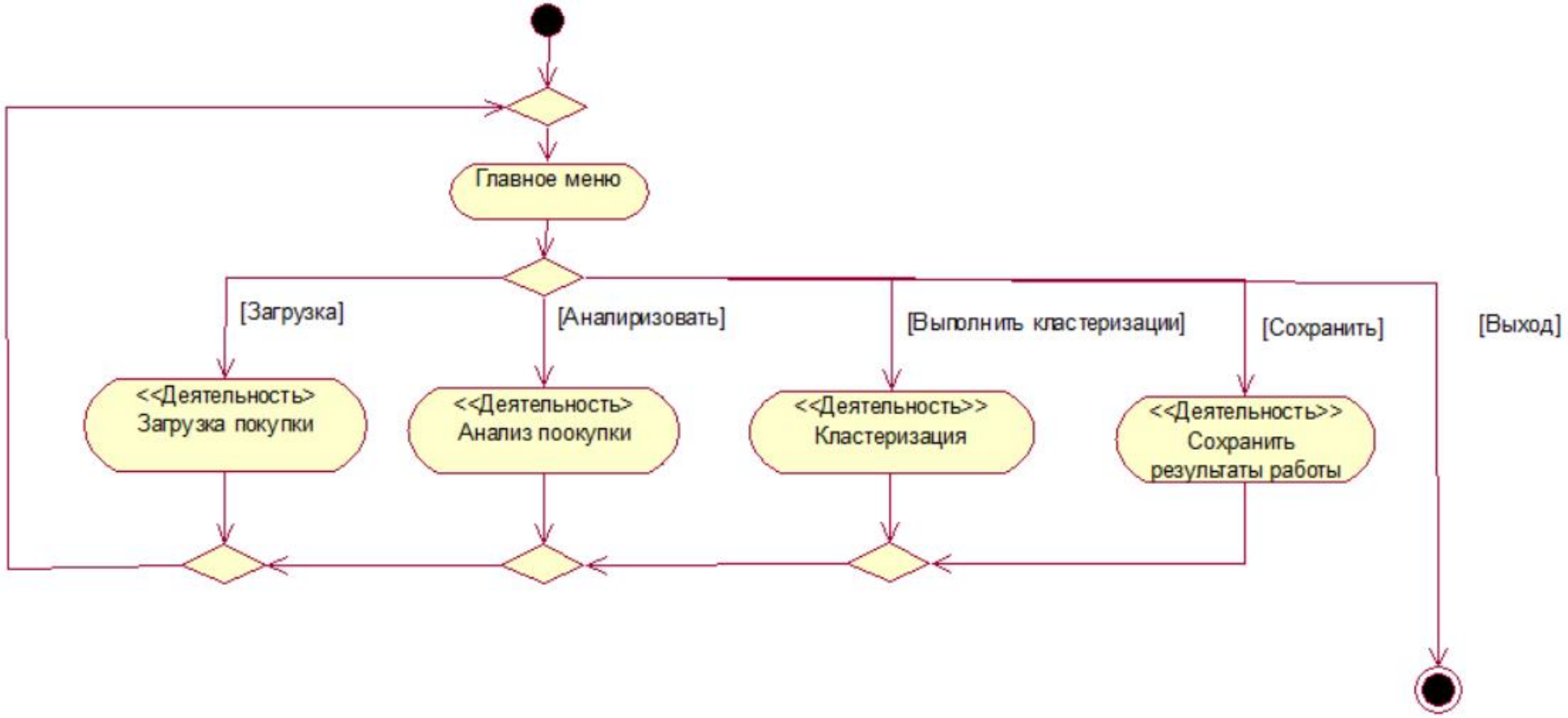



Рисунок 6 – Диаграмма деятельностей «Деятельность алгоритм работы системы»

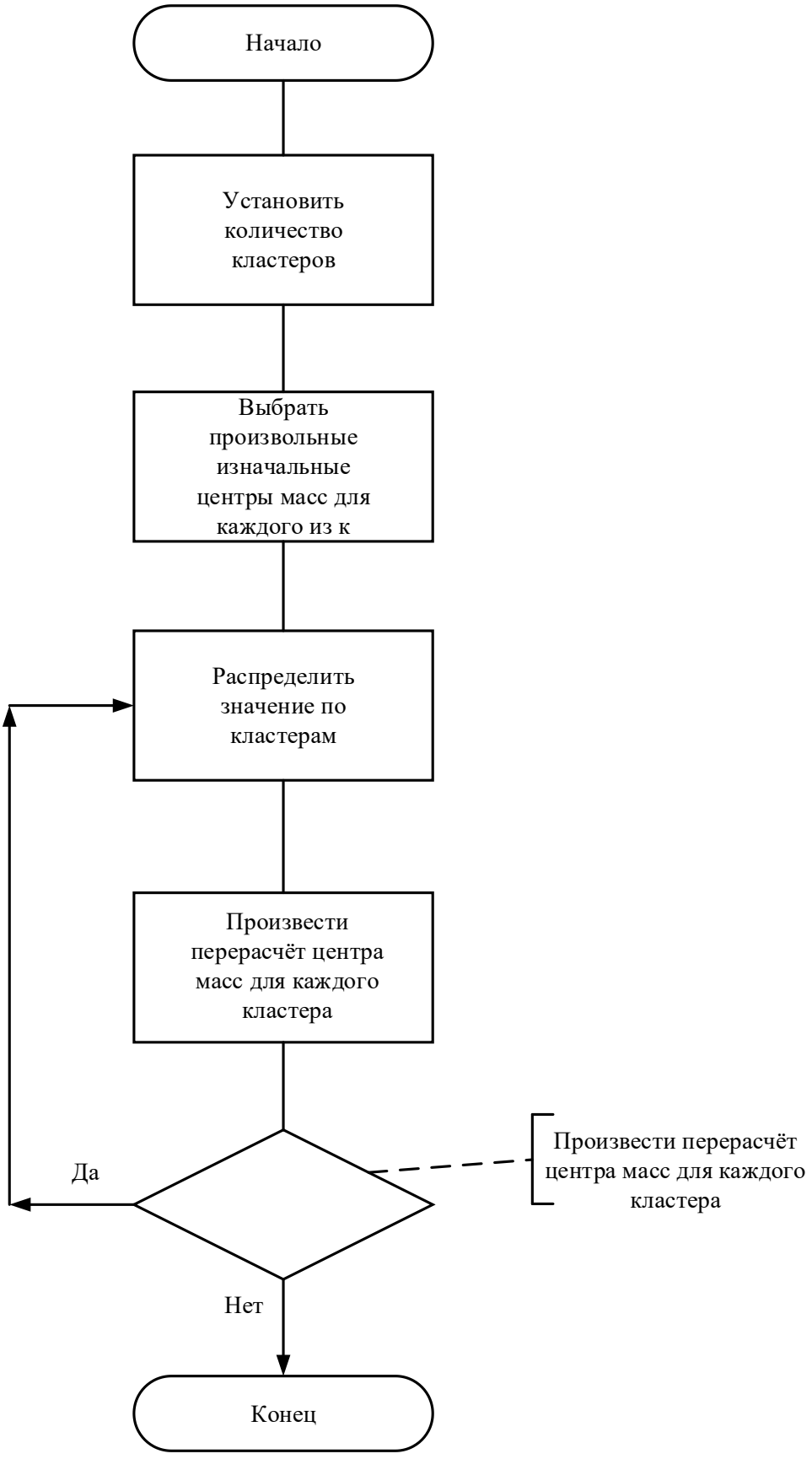



Рисунок 7 – Диаграмма деятельностей «Деятельность кластеризация»

## 2.4 Проектирование интерфейса пользователя

На основании алгоритма функционирования и требований к интерфейсу (см. раздел 1) разработана диаграмма состояний представлен на рисунке 8 и рисунке В.8.

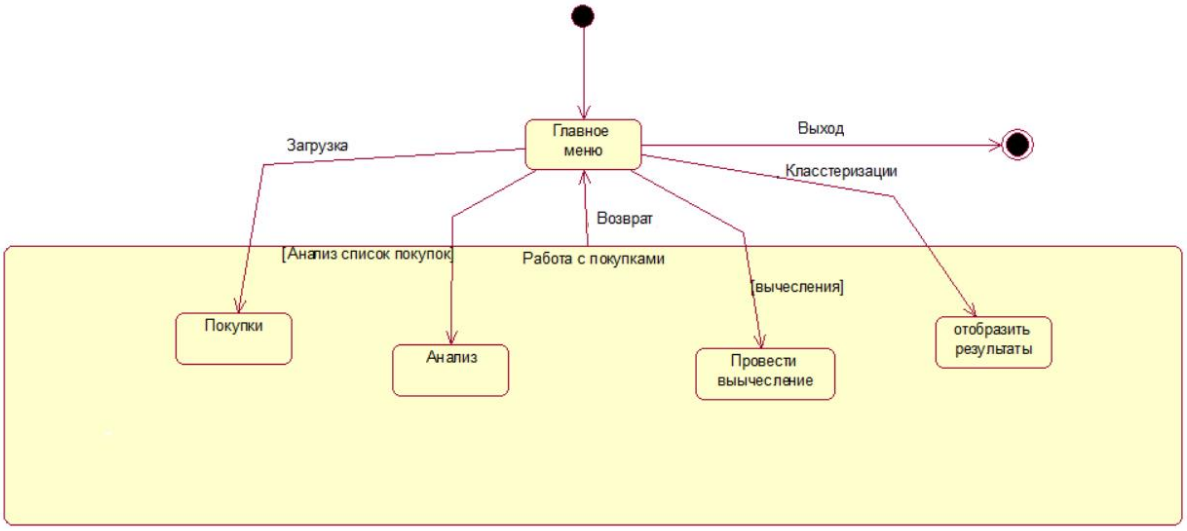



Рисунок 8 – Диаграмма состояний интерфейса

## 2.5 Реляционная модель данных

На рисунке 9 и рисунке В.9 изображена реляционная модель данных.

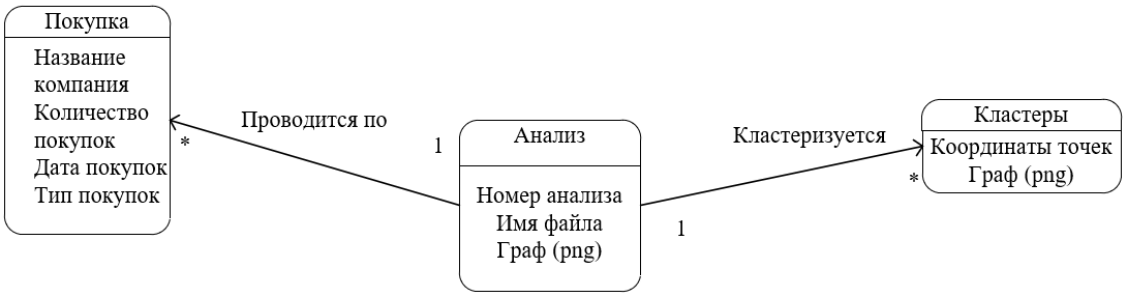



Рисунок 9 – Реляционная модель данных

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 10 и рисунок В.7.

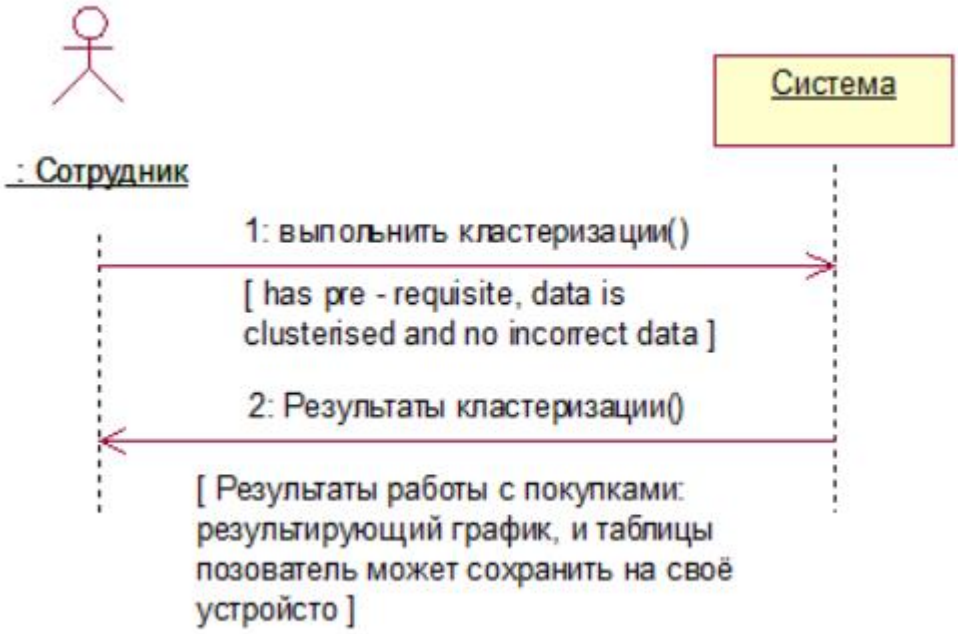



Рисунок 10 – Диаграмма последовательности системных операций варианта использования «выполнить кластеризации»

### 2.6.2 Построение диаграммы кооперации

Структурные особенности передачи и приема сообщений между объектами представлены на диаграмме кооперации на рисунке 11 и рисунке В.6.

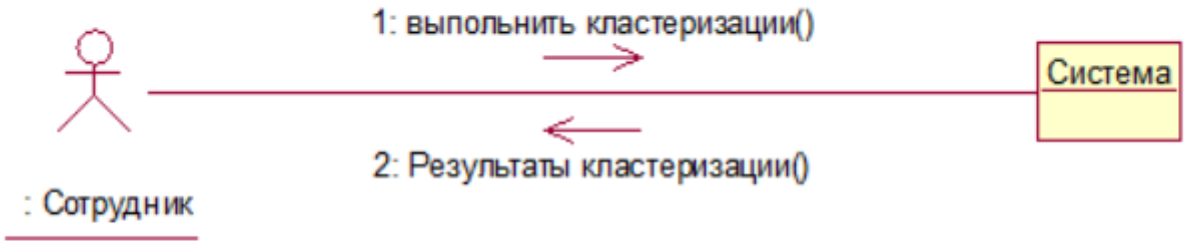



Рисунок 11 – Диаграмма кооперации

# 3 Реализация системы

## 3.1 Реализация программного обеспечения системы

### 3.1.1 Разработка диаграммы компонентов

Реализация программного обеспечения системы представлена на рисунке 12 и рисунке В.5 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 13.

Программный код части разработанной системы, написанной на языке питон с использованием QtDesigner, представлен в приложении А.

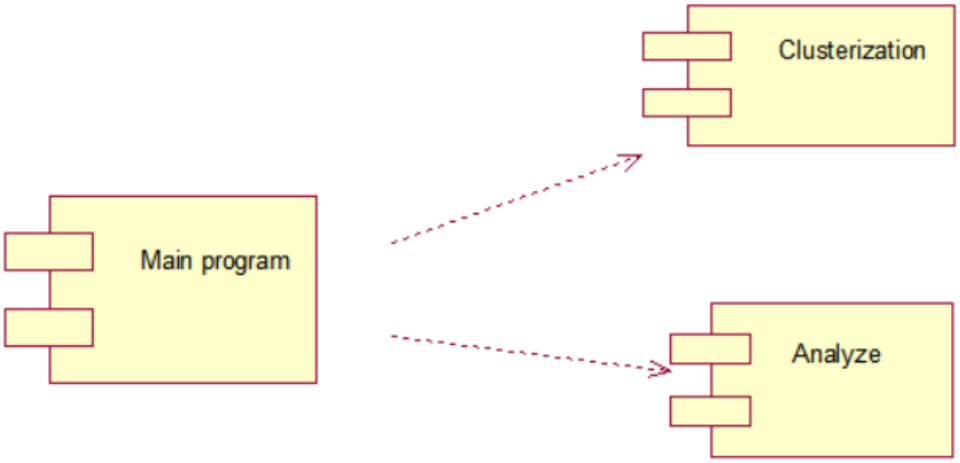



Рисунок 12 – Диаграмма компонентов приложения

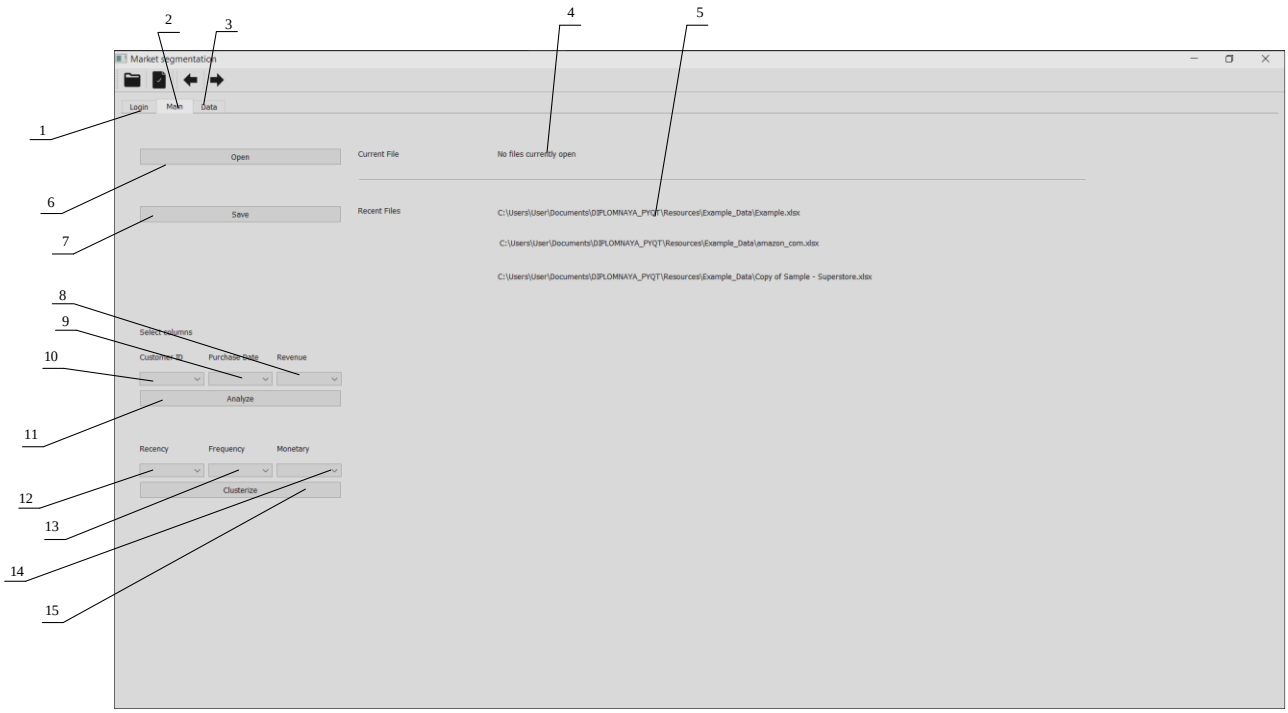



Рисунок 13 – Главная форма

В таблице 5 представлены расположенные на форме MAIN компоненты.

Таблица 5 – Компоненты формы MAIN

| № | Наименование компонента | Тип компо-нента | Назначение |
|---|---|---|---|
| 0 | menuStrip1 | MenuStrip | Панель инструментов |
| 1 | Tab1 | TableWidget1 | Логин и регистрация |
| 2 | Tab2 | TableWidget1 | Главная страница |
| 3 | Tab3 | TableWidget1 | Для отображение загруженных ых |
| 4 | Tlabel1 | Tlabel | Текущий открытый файл |
| 5 | Tlabel2<br>Tlabel3<br>Tlabel4 | Tlabel | Недавно открытые файлы |
| 6 | TButton1 | TButton | Загрузить файл |
| 7 | TButton2 | TButton | Сохранить файл |

Продолжение таблицы 5

| № | Наименование компонента | Тип компо-нента | Назначение |
|---|---|---|---|
| 8 | Combobox1 | TCombobox | Выбрать столбец id покупателей |
| 9 | Combobox2 | TCombobox | Выбрать столбец дата покупок |
| 10 | Combobox3 | TCombobox | Выбрать столбец доходов |
| 11 | TButton3 | TButton | Выполнить анализ |
| 12 | Combobox4 | TCombobox | Выбрать число recency |
| 13 | Combobox5 | TCombobox | Выбрать число frequency |
| 14 | Combobox6 | TCombobox | Выбрать число Monetary |
| 15 | TButton4 | TButton | Выполнить кластеризации |

# 4 Тестирование и оценка трудоемкости разработки программного продукта

## 4.1 Тестирование программного продукта

### 4.1.1 Функциональное тестирование

На основе функциональных требований (1.3.2) могут быть сформулированы следующие тест-требования.

Тест-требования

1. Проверить, что разработанный программный продукт обеспечивает формирование групп на основе кластеризации.

2. Проверить, что разработанный программный продукт обеспечивает доступа к истории покупок клиентов.

3. Проверить, что разработанный программный продукт обеспечивает просмотра различных кластеризованных групп с целью формирования предложений для рекламы.

Результаты функционального тестирования представлены в таблице 6.

Таблица 6 – Результаты тестирования пользовательского интерфейса

| Номер тестового примера | Номер соответствующего тестового требования | Сценарий выполнения тестового примера (действие оператора) | Реакция системы | Результат (пройден / не пройден) |
|---|---|---|---|---|
| 1 | 1 | Откройте файл покупки и выполняете кластеризации | Появляется граф гриппов | Пройден |
| 2 | 2 | Нажмите на кнопку (Open Project) | Откроется окно сохранённых файлов | Пройден |

Продолжение таблицы 6

| Номер тестового примера | Номер соответствующего тестового требования | Сценарий выполнения тестового примера (действие оператора) | Реакция системы | Результат (пройден / не пройден) |
|---|---|---|---|---|
| 3 | 3 | Выполняете кластеризации | Появляется 4 графы с которыми можно создавать предложения покупателей | Пройден |

### 4.1.2 Тестирование пользовательского интерфейса

На основе требований к интерфейсу (1.3.5) могут быть сформулированы следующие тест-требования.

Тест-требования

1. Проверить, что главное меню программного продукта позволяет пользователю выбрать данные о покупателе.

2. Проверить, что главное меню программного продукта позволяет пользователю выполнить кластеризацию данных покупателей.

Результаты тестирования пользовательского интерфейса представлены в таблице 7.

Таблица 7 – Результаты тестирования пользовательского интерфейса

| Номер тестового примера | Номер соответствующего тестового требования | Сценарий выполнения тестового примера (действие оператора) | Реакция системы | Результат (пройден / не пройден) |
|---|---|---|---|---|
| 1 | 1 | Откройте файл покупки | Появляется таблица с данными покупателей | пройден |

Продолжение таблицы 7

| Номер тестового примера | Номер соответствующего тестового требования | Сценарий выполнения тестового примера (действие оператора) | Реакция системы | Результат (пройден / не пройден) |
|---|---|---|---|---|
| 2 | 2 | Нажмите на кнопку (Analyze) и (Clusterize) | Появляется несколько графов | пройден |

### 4.1.3 Модульное тестирование

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.

Модульные тесты представлены в приложении Б.

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 14).

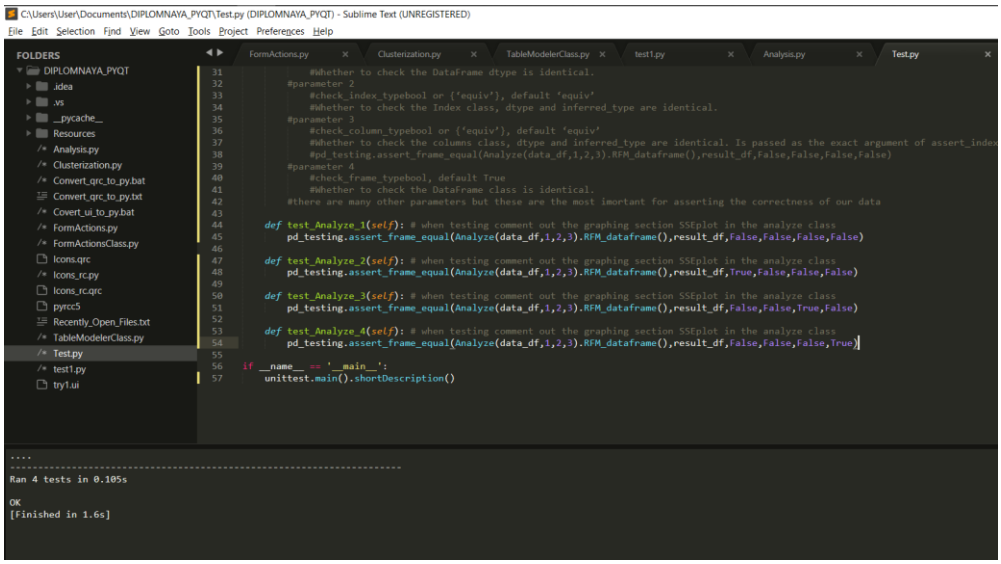



Рисунок 14 – Вывод успешно выполненного теста

## 4.2 Оценка трудоемкости создания программного продукта

Для разработанного программного продукта был выбран метод оценки трудоемкости разработки программных продуктов на основе функциональных указателей. Данный метод позволяет наиболее точно оценивать т. к. полученные результаты основаны на информационные характеристики компоненты.

Исходные данные.

Функции разрабатываемой системы:

– формирование групп на основе кластеризации,

– обеспечение доступа к истории покупок клиентов,

– обеспечение просмотра различных кластеризованных групп с целью.

Ввод и накопление информации: данные в виде текста программы вводятся с клавиатуры, настройки системы производятся пользователем с помощью форм программы. Информация, относящаяся к системе, хранится в базе данных.

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

Выделенные согласно методике информационные характеристики показаны в таблице 8.

Таблица 8 – Информационные характеристики

| Наименование | Число элементов данных | Число ссылок на файлы / Типы элементов записей | Ранг |
|---|---|---|---|
| *Внешние вводы:* | | | |
| 1. Форма главная | 13 | 1 | 3 |
| 2. Форма управления | 4 | 0 | 3 |
| *Внешние выводы:* | | | |
| 1. Форма результатов анализа | 10 | 0 | 4 |
| *Внутренние логические файлы:* | | | |
| 1. Форма таблиц | 1 | 1 | 7 |
| *Общее количество рангов* | | | 17 |

Значения системных параметров приложения представлены в таблице 9.

Таблица 9 – Значения системных параметров приложения

| № | Системный параметр | Значение (Fi) |
|---|---|---|
| 1 | Передача данных | 0 |
| 2 | Распределенная обработка данных | 0 |
| 3 | Производительность обработки | 4 |

Продолжение таблицы 9

| № | Системный параметр | Значение (Fi) |
|---|---|---|
| 4 | Эксплуатационные ограничения | 0 |
| 5 | Частота транзакций | 0 |
| 6 | Ввод данных в режиме «онлайн» | 0 |
| 7 | Эффективность работы | 1 |
| 8 | Онлайновое обновление | 0 |
| 9 | Сложная обработка | 1 |
| 10 | Повторное использование | 1 |
| 11 | Простота установки | 0 |
| 12 | Простота эксплуатации | 0 |
| 13 | Количество возможных установок на различных платформах | 1 |
| 14 | Простота изменений (гибкость) | 1 |
| Итого (Σ Fi) | | 9 |

Количество функциональных указателей вычисляется по формуле (1) с учетом данных таблицы 8 и 9:

$$FP = \text{Общее количество рангов} \times (0{,}65 + 0{,}01 \times \sum Fi) = 12{,}58 \quad (1)$$

Полученная FP-оценка пересчитывается в LOC-оценку V, учитывая,что ПП создается с использованием среды питон.

$$V = K_{яз} \times FP = 12{,}58. \times 55 = 0{,}692 \text{ KSLOC} \quad (2)$$

Разрабатываемое ПО относится в полунезависимому, поэтому берёмсоответствующие коэффициенты N1, N2, N3.

Согласно формуле $T = N1 \times KSLOC^{N2}$ трудоемкость создания ПП составляет

$$T = 3{,}0 \times 0{,}692^{1{,}12} = 1{,}99 \text{ чел.} \cdot \text{ мес} \quad (3)$$

Время разработки ПП составляет согласно формуле (4) составляет

$$t_{разр} = 2{,}5 \times T^{N3} = 3{,}2 \text{ мес.} \quad (4)$$

Результаты расчета:

1) трудозатраты на разработку проекта составят: 1,99 чел.·мес,
2) время разработки составит: 3,2 мес.

Согласно полученным результатам вычислений трудоёмкости, данный проект может быть реализован командой разработчиков за срок 3,2 месяца в то же время реальное время разработки не может превышать (или составило) 9 месяцев разработчиком одиночкой, что позволяет сделать вывод о том, что оценка трудоёмкости проведена с достаточной точностью.

# ЗАКЛЮЧЕНИЕ

Разработанный программный продукт позволяет автоматизировать процесс кластеризации пользователей продуктового магазина. Этот программный продукт упрощает процесс прогнозирования рекламы, программный продукт может использоваться в качестве подсистемы в приложении для сокращения количества приложений.

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. Была выполнена частично проверка и отладка системы.

Система позволяет осуществить:

– формирование групп на основе кластеризации,

– доступ к истории покупок клиентов,

– просмотр различных кластеризованных групп с целью формирования предложений для рекламы.

В данной выпускной квалификационной работе было проведено программирование на языке Python. Главное меню, события кнопок, визуальная составляющая проекта прописаны в приложения qt Designer, функции выполняющий кластеризации были прописаны на языке Python 3.9. Главное меню программного продукта позволяет выбрать файл, содержащий данных покупателей, выполнит анализ и выбрать значение на кластеризации и выполнит кластеризации над данными.

Таким образом, в бакалаврской работе удалось реализовать интеллектуальную систему кластеризации для продуктового магазина, которая удовлетворяет заданным требованиям.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1. Обзор алгоритмов кластеризации [Сайт]. URL: https://scikit-learn.org/stable/modules/clustering.html c/ (дата обращения: 01.10.2020).
2. Интерфейс на питон [Сайт]. URL: / Обзор алгоритмов кластеризации [Сайт]. URL: https://www.riverbankcomputing.com/static/Docs/PyQt5/ (дата обращения: 01.10.2020).
3. Обзор алгоритмов кластеризации [Сайт]. URL: https://scikit-learn.org/stable/modules/clustering.html/ (дата обращения: 01.03.2021).
4. RFM алгоритмов [Сайт]. 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 [Сайт]. 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 [Сайт]. URL: https://bl.ocks.org/rpgove/0060ff3b656618e9136b/ (дата обращения: 01.11.2020).
7. Наборы данных [Сайт]. URL: https://www.kaggle.com/vijayuv/onlineretail/ (дата обращения: 01.02.2021).
8. RFM-анализ для сегментации клиентов и маркетинга лояльности [Сайт]. URL: https://www.youtube.com/watch?v=OYohJxp2l9k/ (дата обращения: 25.01.2021).
9. RFM-анализ для сегментации клиентов и маркетинга лояльности [Сайт]. URL: https://www.ibm.com/docs/en/spss-statistics/27.0.0?topic=marketing-direct/ (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 purchas-
ers 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_clus-
ter(freq_k, 'Frequency', self.rfm_df)
                                    self.rfm_df.groupby('Frequen-
cyCluster')['Frequency']

                                    self.rfm_df  =  self.recency_clus-
ter(recen_k, 'Recency', self.rfm_df)
                                    self.rfm_df.groupby('RecencyClus-
ter')['Recency']

                                    self.rfm_df  =  self.revenue_clus-
ter(rev_k, 'logRevenue', self.rfm_df)
                                    self.rfm_df.groupby('RevenueClus-
ter')['logRevenue']

                                    self.rfm_df['OverallScore']     =
self.rfm_df['RecencyCluster']  +  self.rfm_df['FrequencyCluster']  +
self.rfm_df['RevenueCluster']
                                    self.rfm_df.groupby('Over-
allScore')['Recency','Frequency','logRevenue'].mean()

                                    self.graphClusters()
                                    #self.graphIndividaulMeasures()
```

```
                    def    order_cluster(self,cluster_col,    fea-
ture_col, df, ascending):
                                df_new     =     df.groupby(clus-
ter_col)[feature_col].mean().reset_index()
                                df_new     =     df_new.sort_val-
ues(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([clus-
ter_col],axis=1)
                                df_final   =   df_final.rename(col-
umns={"index":cluster_col})

                                return df_final

                    def   frequency_cluster(self,   cluster_number,
frequency_col, dataframe):
                                frequency_kmeans = KMeans(n_clus-
ters=cluster_number)
                                frequency_kmeans.fit(data-
frame[[frequency_col]])

                    # Assigning cluster prediction to customers
                                dataframe['FrequencyCluster']    =
frequency_kmeans.predict(dataframe[[frequency_col]])

                    # Ordering clusters from low to high and iden-
tifying statistics
                                dataframe    =    self.order_clus-
ter('FrequencyCluster', frequency_col, dataframe, True)

                                return dataframe
```

```
                    def recency_cluster(self, cluster_number, re-
cency_col, dataframe):
                                recency_kmeans = KMeans(n_clus-
ters=cluster_number)
                                recency_kmeans.fit(dataframe[[re-
cency_col]])

                    # Assigning cluster prediction to customers
                                dataframe['RecencyCluster'] = re-
cency_kmeans.predict(dataframe[[recency_col]])

                    # Ordering clusters from low to high and iden-
tifying statistics
                                dataframe = self.order_clus-
ter('RecencyCluster', recency_col, dataframe, False)

                                return dataframe

                    def revenue_cluster(self, cluster_number, rev-
enue_col, dataframe):
                                revenue_kmeans = KMeans(n_clus-
ters=cluster_number)
                                revenue_kmeans.fit(data-
frame[[revenue_col]])

                    # Assigning cluster prediction to customers
                                dataframe['RevenueCluster'] = rev-
enue_kmeans.predict(dataframe[[revenue_col]])

                    # Ordering clusters from low to high and iden-
tifying statistics
                                dataframe = self.order_clus-
ter('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['Over-
allScore']>4,'Segment'] = 1

                    self.rfm_df.loc[self.rfm_df['Over-
allScore']>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, projec-
tion='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  Representa-
tion of Segments', loc='left')
                                    plt.show();


                        def graphIndividaulMeasures(self):
                                    plt.figure()
                                    plt.scatter(list(self.rfm_df['Re-
cencyCluster']))
                                    plt.show();
```

## ПРИЛОЖЕНИЕ Б

### Программный код модульных тестов

```
import unittest
from  Clusterization import Clusterize
from Analysis import Analyze
import pandas as pd
import pandas.testing as pd_testing
from matplotlib.testing.decorators import image_comparison

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']
        }

data_df  =  pd.DataFrame  (data,  columns  =  ['Order  ID','Order
Date','Profit'])

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]
        }

result_df = pd.DataFrame (result, columns = ['Order ID','Frequen-
cy','Recency','logRevenue'])

class TestApp(unittest.TestCase):

                    #Analyze(data_df,1,2,3).RFM_dataframe()
```

```
                    " when testing comment out the graphing section
SSEplot in the analyze class"
                    #Analyze assert_frame_equal parameters:
                                #parameter 1
                                                #check_dtypebool,
default True
                                                #Whether to check
the DataFrame dtype is identical.
                                #parameter 2
                                                #check_in-
dex_typebool or {'equiv'}, default 'equiv'
                                                #Whether to check
the Index class, dtype and inferred_type are identical.
                                #parameter 3
                                                #check_col-
umn_typebool or {'equiv'}, default 'equiv'
                                                #Whether to check
the columns class, dtype and inferred_type are identical. Is passed
as the exact argument of assert_index_equal().
                                                #pd_testing.as-
sert_frame_equal(Analyze(data_df,1,2,3).RFM_dataframe(),re-
sult_df,False,False,False,False)
                                #parameter 4

                    #check_frame_typebool, default True
                                                #Whether to check
the DataFrame class is identical.
                                #there are many other parameters
but these are the most imortant for asserting the correctness of our
data

                    def test_Analyze_1(self): # when testing com-
ment out the graphing section SSEplot in the analyze class
```

```
                                    pd_testing.assert_frame_equal(An-
alyze(data_df,1,2,3).RFM_dataframe(),re-
sult_df,False,False,False,False)

                    def test_Analyze_2(self): # when testing com-
ment out the graphing section SSEplot in the analyze class
                                    pd_testing.assert_frame_equal(An-
alyze(data_df,1,2,3).RFM_dataframe(),re-
sult_df,True,False,False,False)

                    def test_Analyze_3(self): # when testing com-
ment out the graphing section SSEplot in the analyze class
                                    pd_testing.assert_frame_equal(An-
alyze(data_df,1,2,3).RFM_dataframe(),re-
sult_df,False,False,True,False)

                    def test_Analyze_4(self): # when testing com-
ment out the graphing section SSEplot in the analyze class
                                    pd_testing.assert_frame_equal(An-
alyze(data_df,1,2,3).RFM_dataframe(),re-
sult_df,False,False,False,True)

if __name__ == '__main__':
    unittest.main().shortDescription()
```

## ПРИЛОЖЕНИЕ В

## Внешний вид графического материала

Внешний вид графического материала, выполненного на отдельных листах представлен на рисунках В.1 – В.10.

Перечень графического материала:

лист 1: Постановка задачи (рисунок В.1),

лист 2: Диаграмма функциональных требования к системе (рисунок В.2),

лист 3: Диаграмма деятельностей «Деятельность алгоритм работы системы» (рисунок В.3),

лист 4: Диаграмма концептуальной модели предметной области (рисунок В.4),

лист 5: Диаграмма компонентов приложения (рисунок В.5),

лист 6: Диаграмма кооперации (рисунок В.6),

лист 7: Диаграмма последовательности системных операций варианта использования «выполнить кластеризации» (рисунок В.7),

лист 8: Диаграмма состояний интерфейса (рисунок В.8),

лист 9: Диаграмма реляционной модели данных (рисунок В.9),

лист 10: Контекстная SADT-диаграмма системы (рисунок В.10).

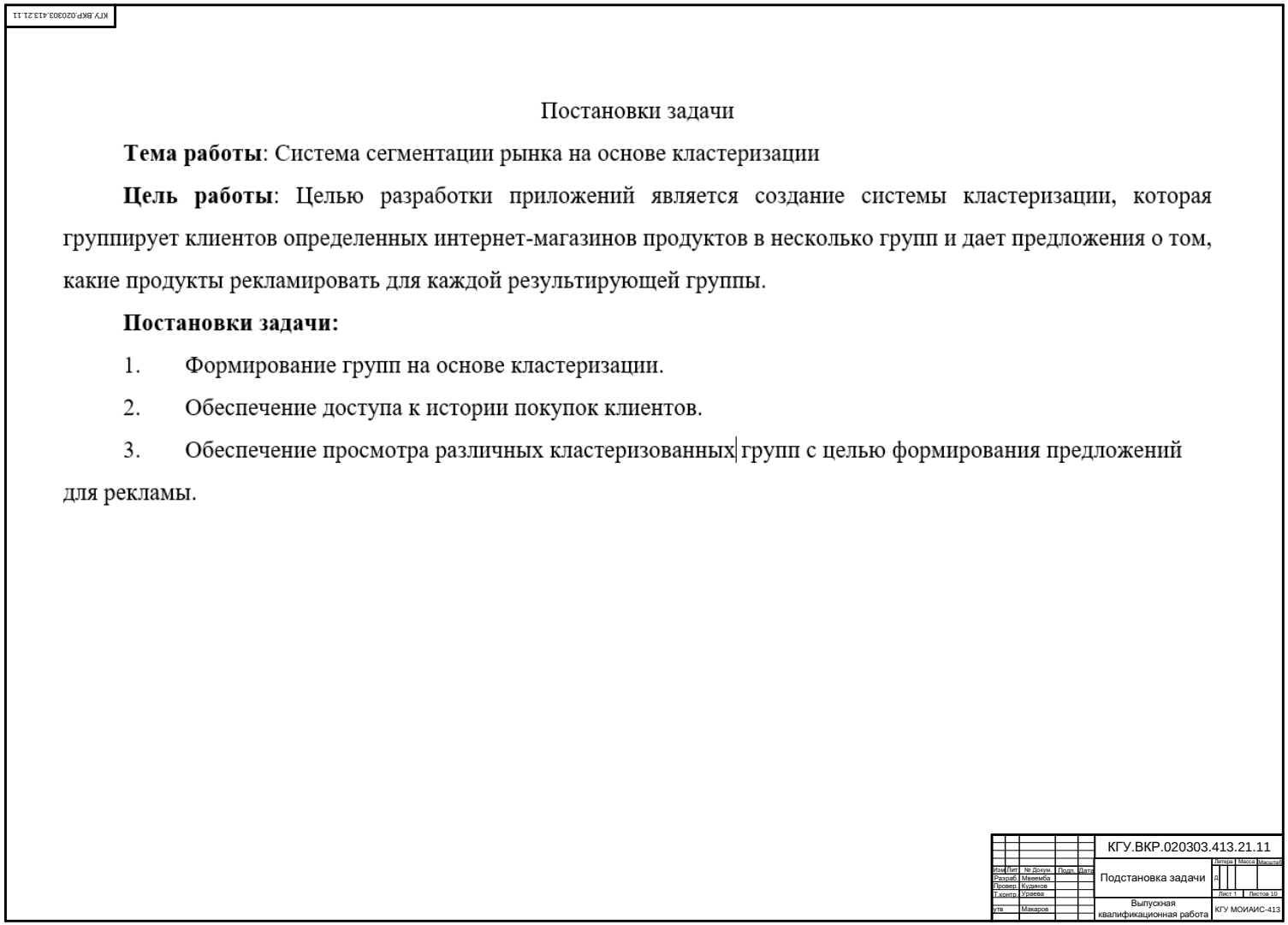



Рисунок В.1 – Постановка задачи

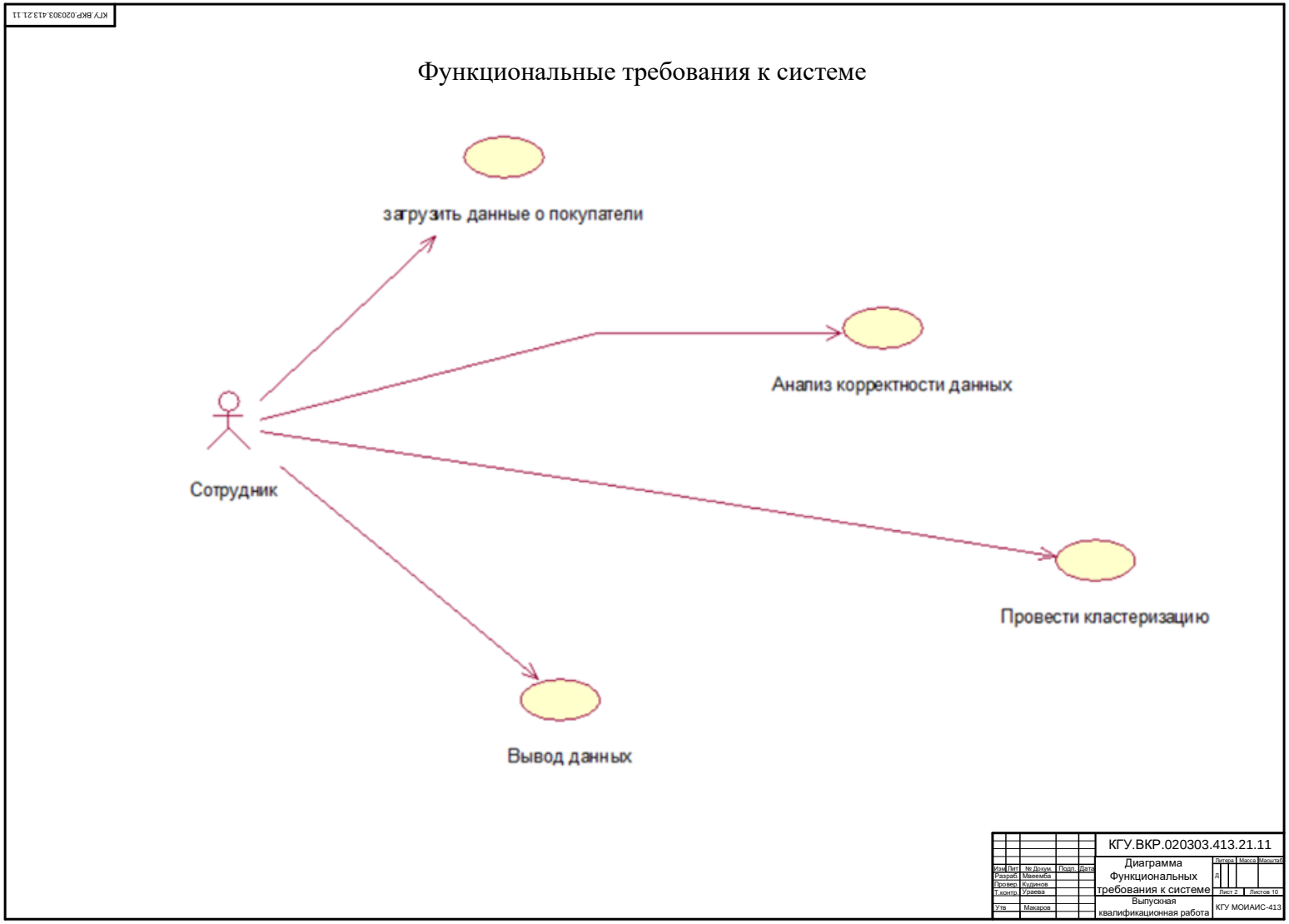



Рисунок В.2 – Диаграмма функциональных требования к системе

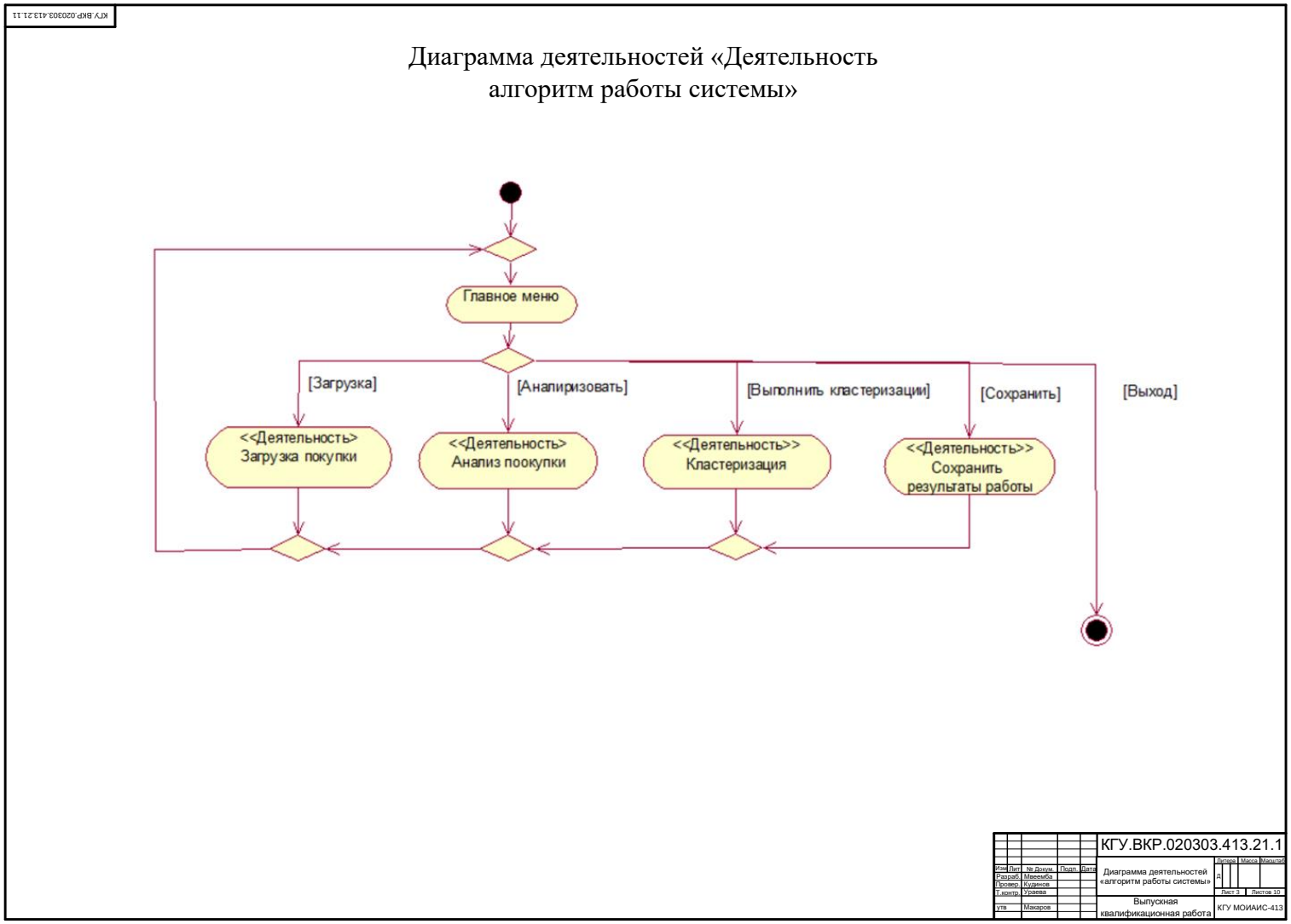



Рисунок В.3 – Диаграмма деятельностей «Деятельность алгоритм работы системы»

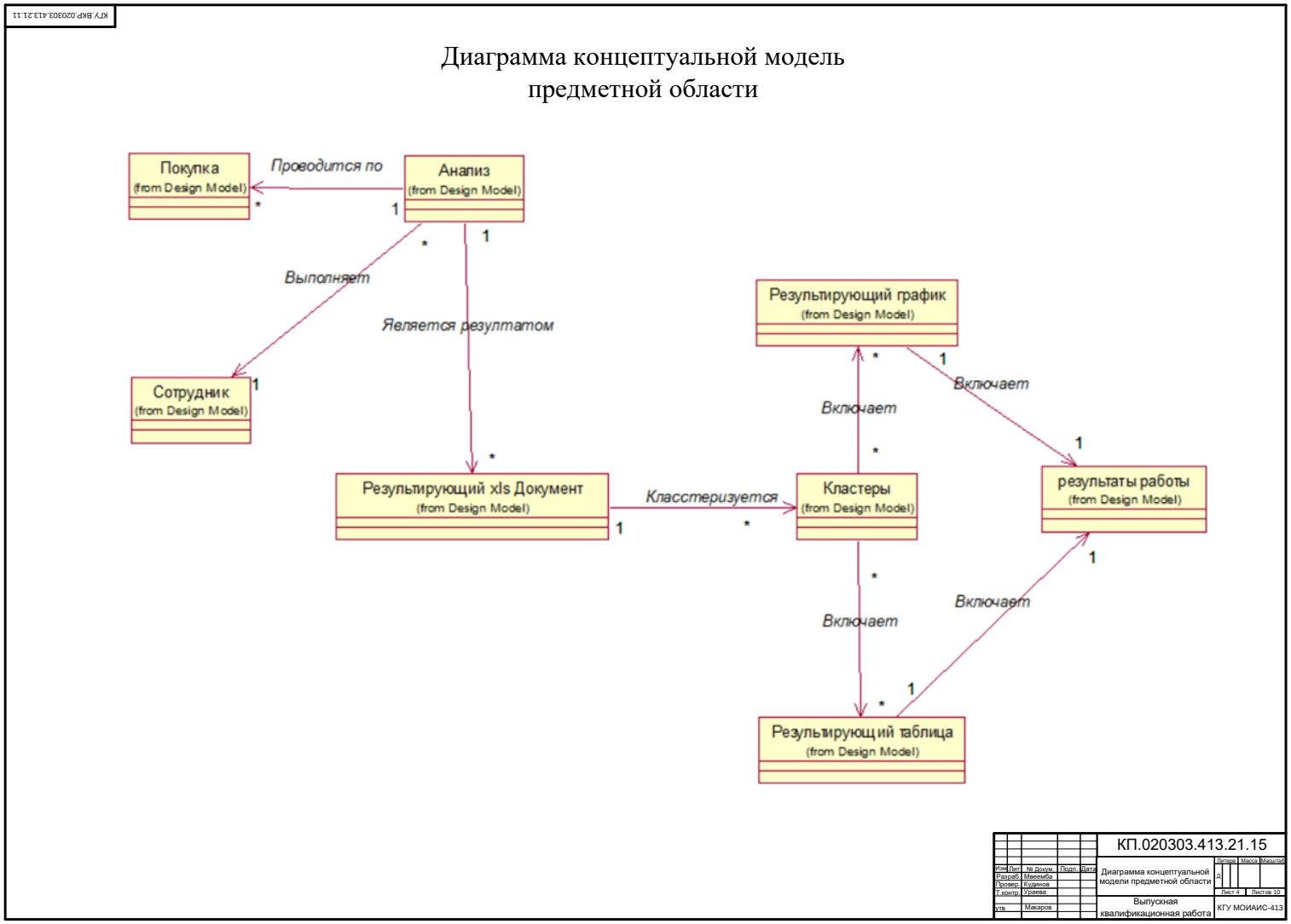



Рисунок В.4 – Диаграмма концептуальной модели предметной области

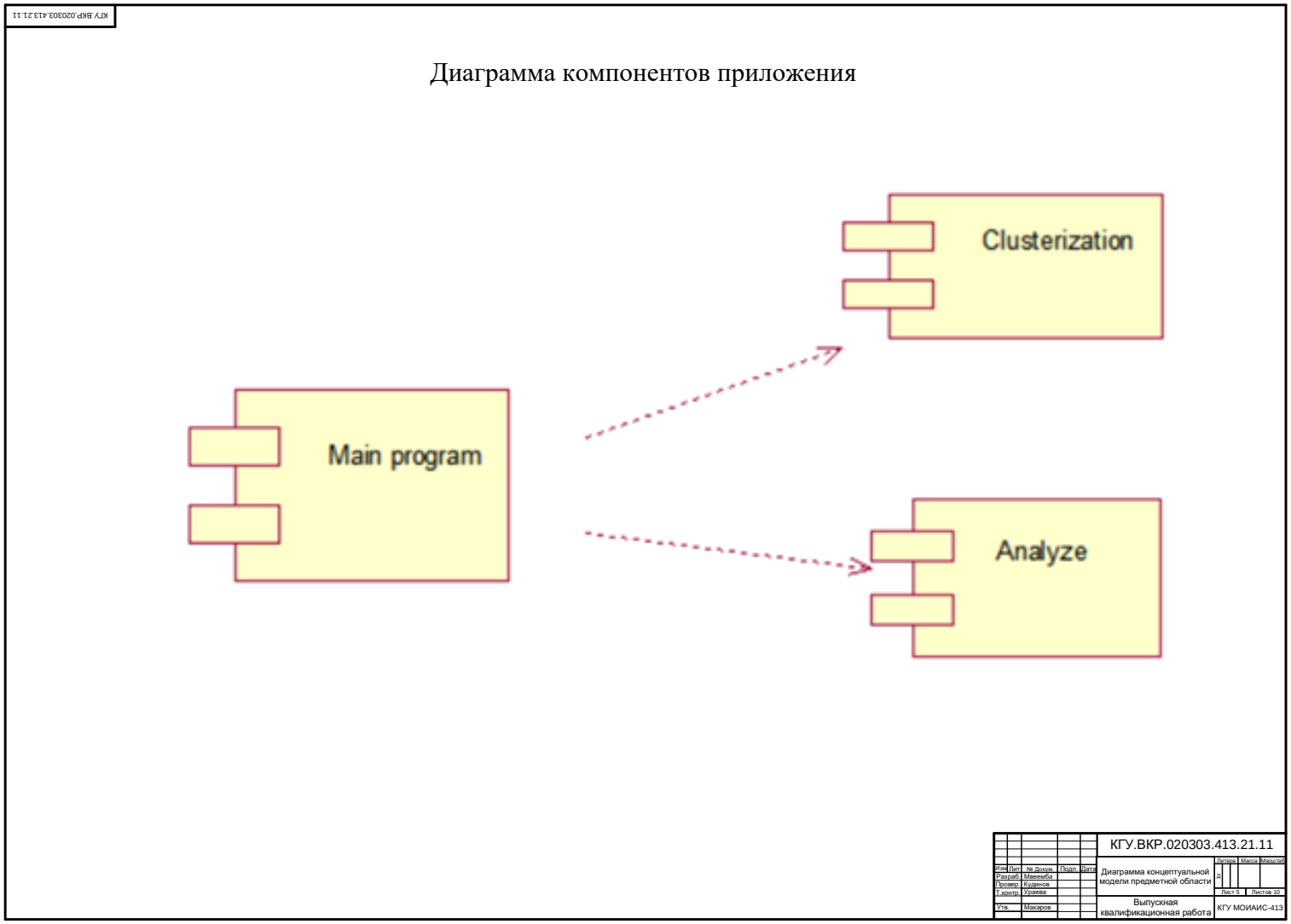



Рисунок В.5 – Диаграмма компонентов приложения

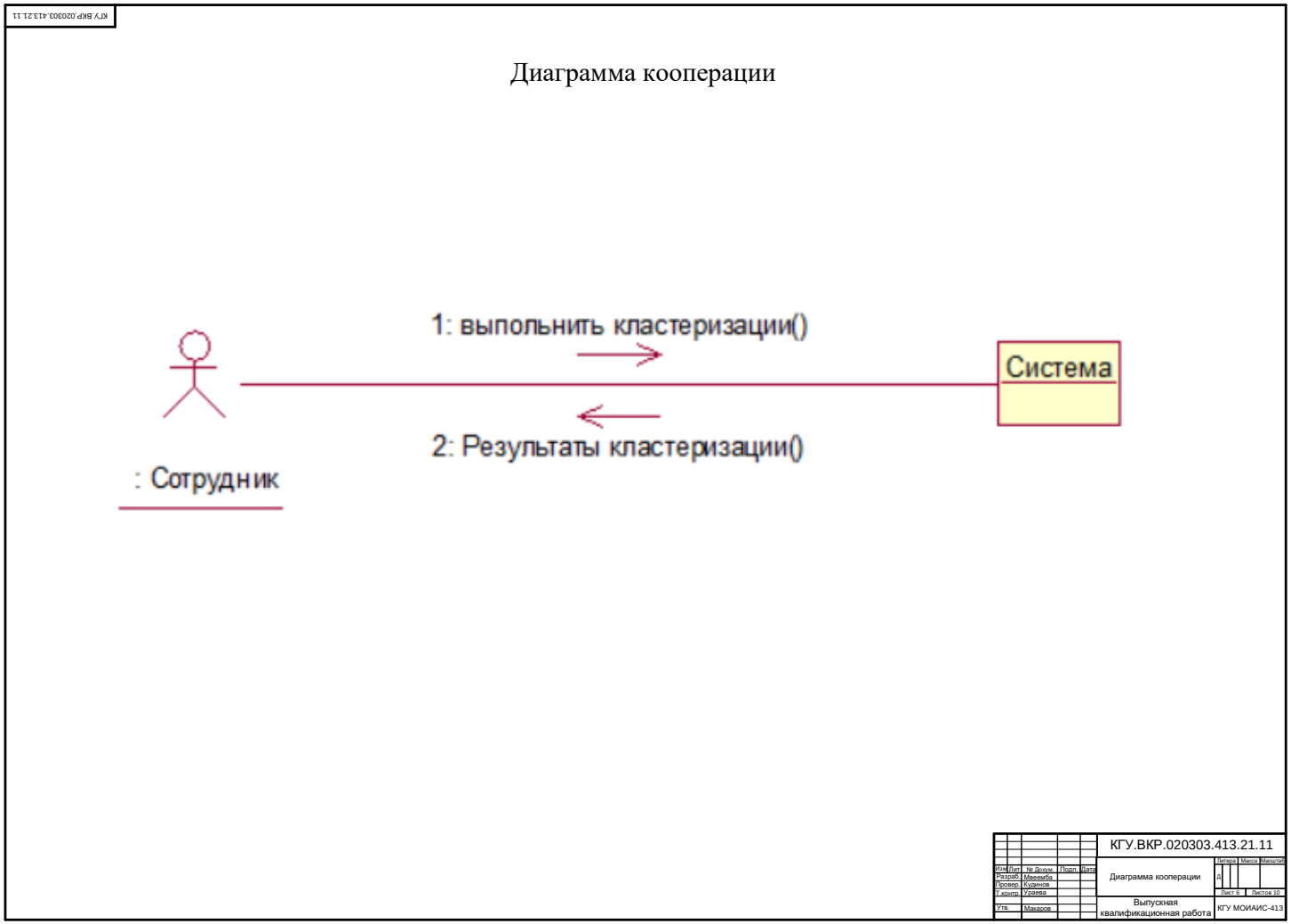



Рисунок В.6 – Диаграмма кооперации

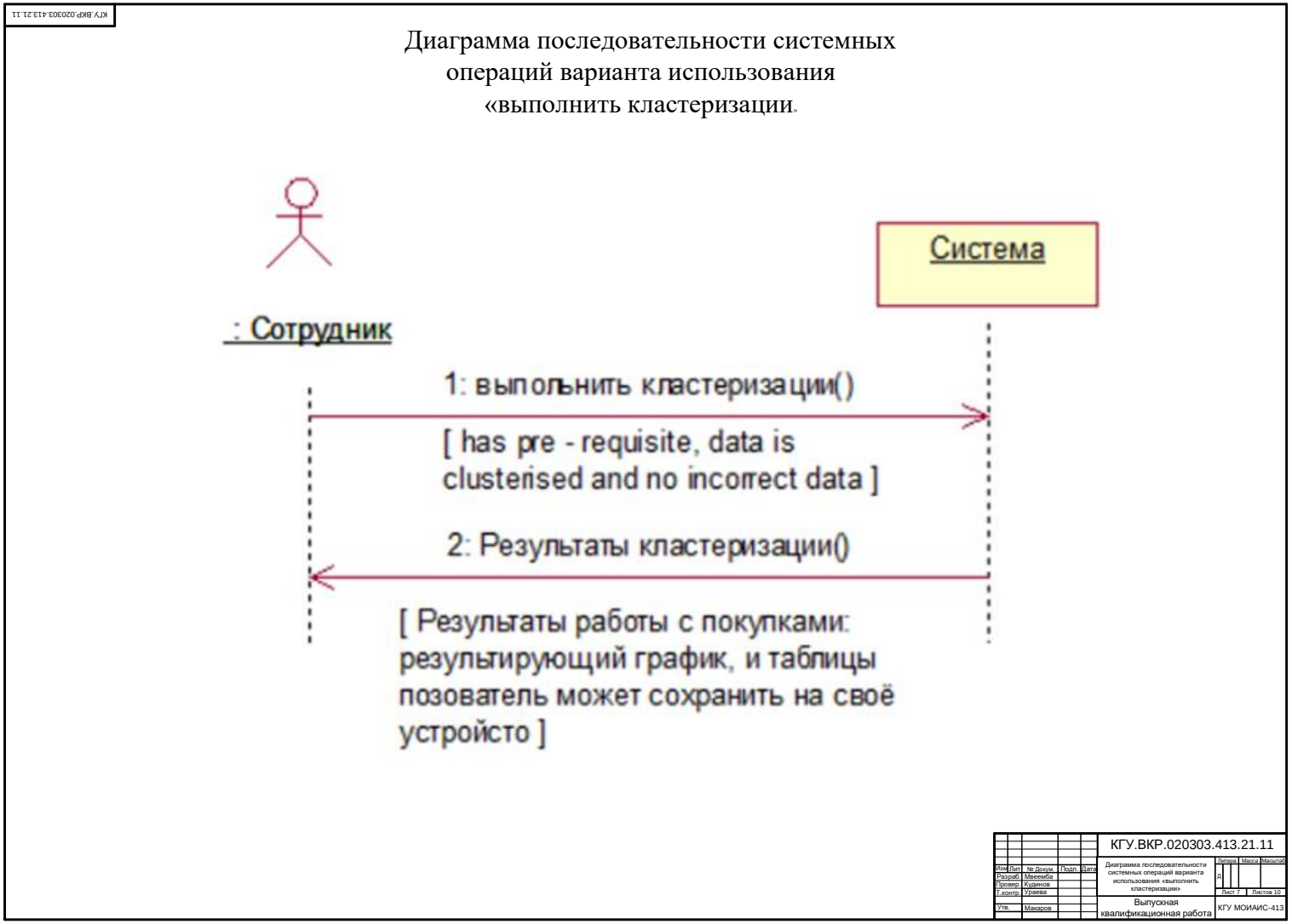



Рисунок В.7 – Диаграмма последовательности системных операций варианта использования «выполнить кластеризации»

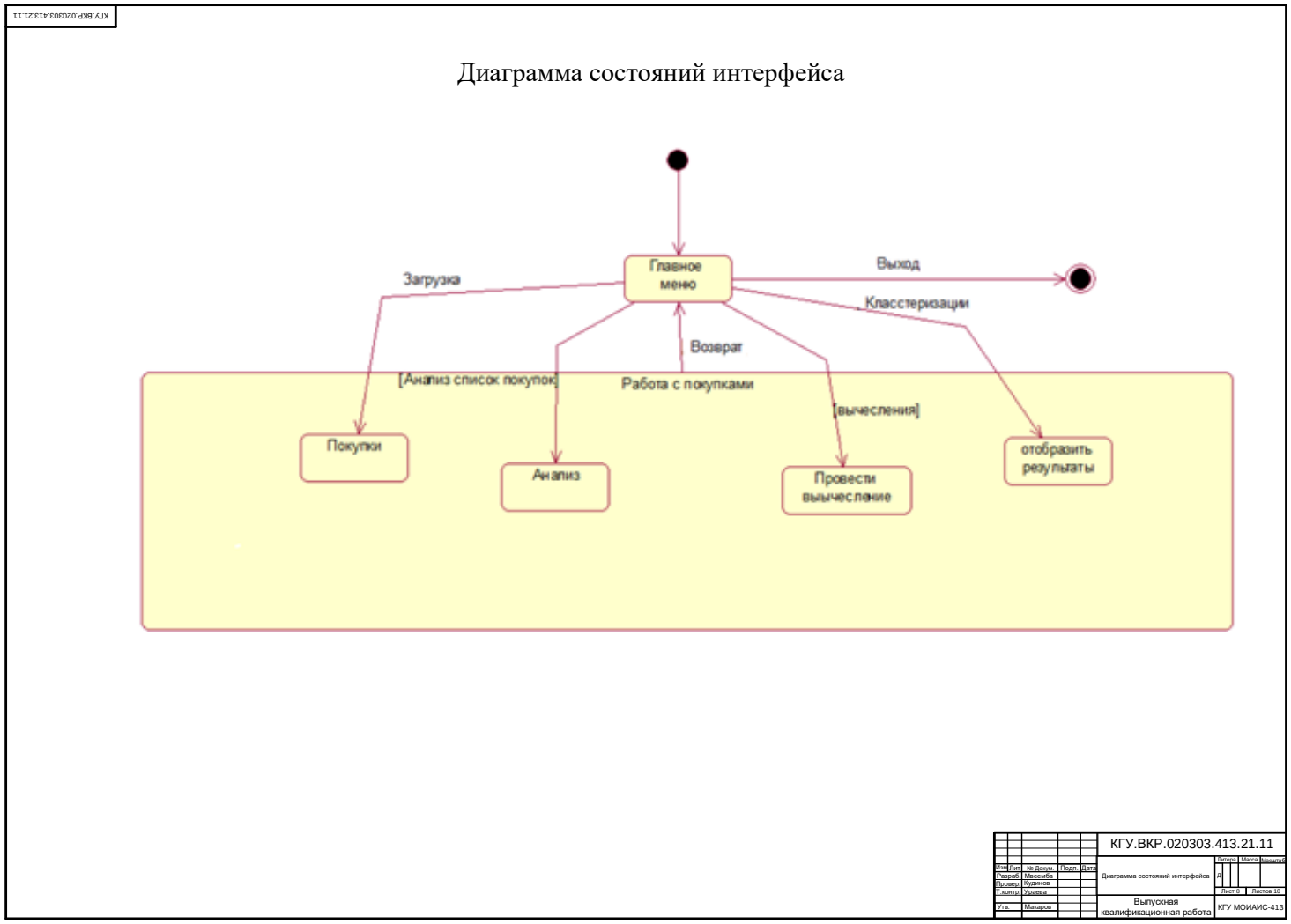



Рисунок В.8 – Диаграмма состояний интерфейса

КГУ.ВКР.020303.413.21.11

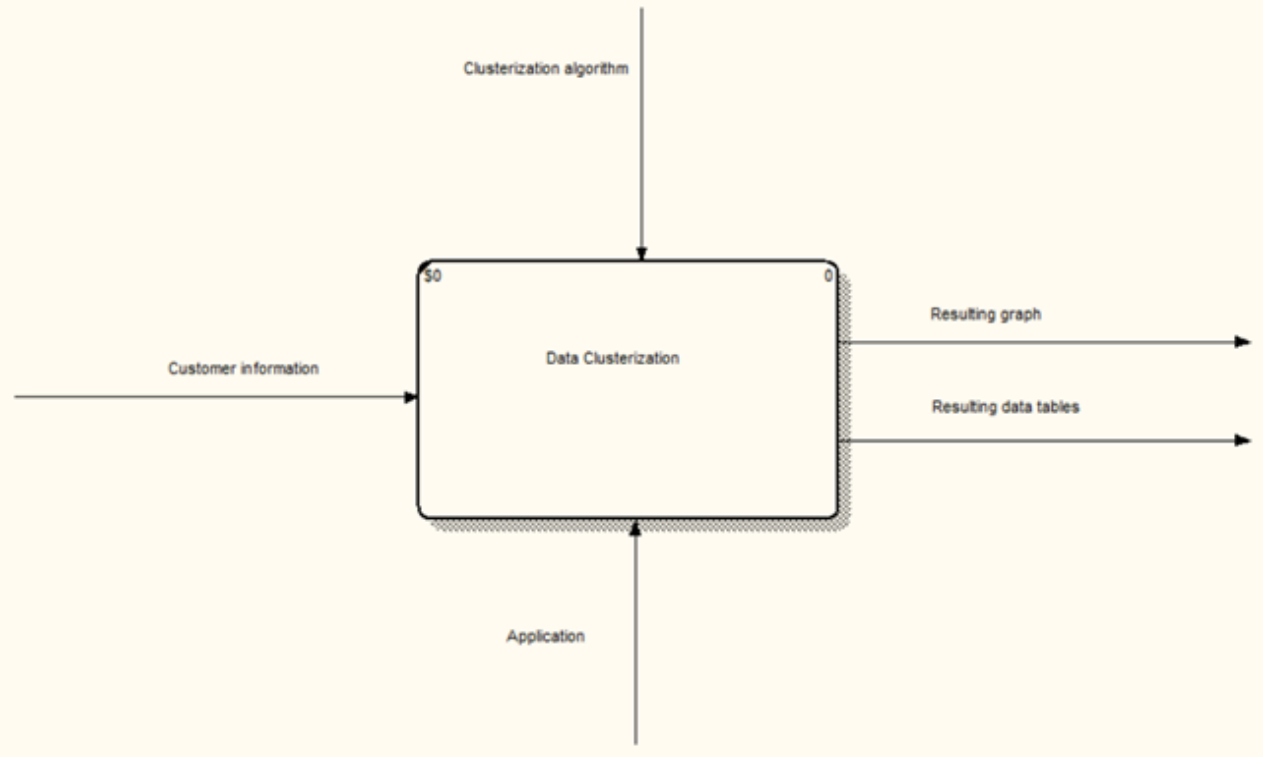
Диаграмма Реляционной модели данных


| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Изм. | Лит. | № докум. | Подп. | Дата | КГУ.ВКР.020303.413.21.11 | | |
| Разраб. | | Мвеемба | | | Диаграмма реляционной модели данных | Литера: А | Масса: | Масштаб: |
| Провер. | | Куренов | | | | Лист 9 | Листов 10 | |
| Т.контр. | | Ураева | | | Выпускная квалификационная работа | КГУ МОИАИС-413 | | |
| Утв. | | Макаров | | | | | | |

Рисунок В.9 – Диаграмма реляционной модели данных

КГУ.ВКР.020303.413.21.11

Контекстная SADT-диаграмма системы




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Изм. | Лит. | № докум. | Подп. | Дата | КГУ.ВКР.020303.413.21.11 | | |
| Разраб. | | Мвеемба | | | Контекстная SADT-диаграмма системы | Литера | Масса / Масштаб |
| Провер. | | Куренов | | | | А | |
| Т.контр. | | Уразва | | | | Лист 10 | Листов 10 |
| Утв. | | Макаров | | | Выпускная квалификационная работа | КГУ МОИАИС-413 | |

Рисунок В.10 – Контекстная SADT-диаграмма системы