

Задание 1

Пользовательские и функциональные требования

Drow.io для рисования use-case-ов -

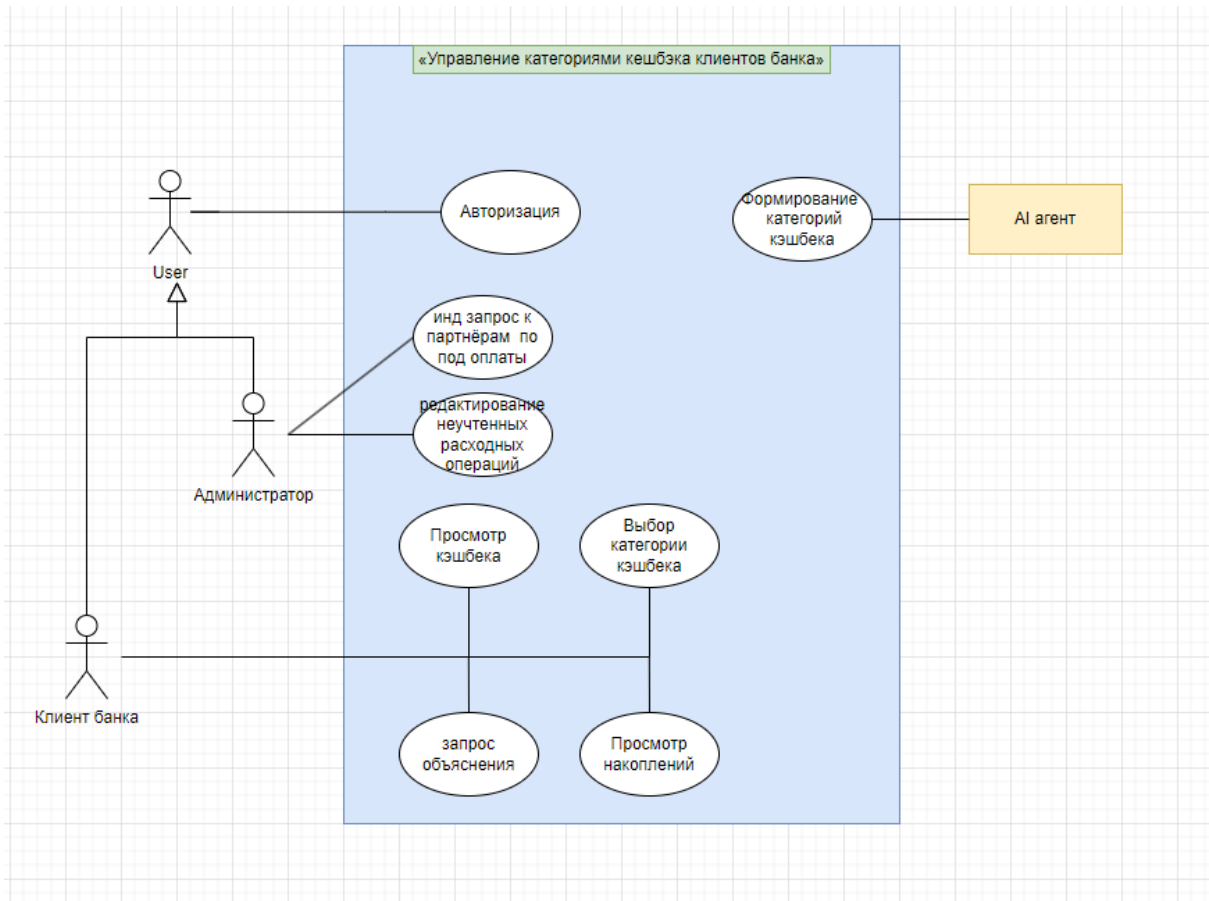
<https://drive.google.com/file/d/18JW7CIkydUi7vzAloOTqmFh88KBTdxvi/view?usp=sharing>

Vision -

<https://docs.google.com/document/d/1NhLUc4sldfSrsprlH0FlfBzeWrs3d1OBNry02HFB6Og/edit?usp=sharing>

Критерии MoSCoW для спецификации требований:

1. **Must have** (обязан иметь) - Обязательное требование, являющиеся фундаментальным для системы
2. **Should have** (должен иметь) - Важные требования, которые могут быть опущены.
3. **Could have** (мог бы иметь) - По-настоящему необязательные требования (реализуются, если есть на это время).
4. **Want to have** (хотелось бы иметь) - Требования, которые могут подождать до следующих версий системы.



Пользовательские и функциональные требования			
Actor	Use case	User story	функциональн ые требования
<Actor>	<Use-case>	Как <сущность>, я хочу <действие>, чтобы <результат>.	<id> <система> <MoSCoW> <действие>
User	авторизация	как пользователь системы, я хочу авторизоваться в системе и успешно переходить в свой личный кабинет	<1> система обязана уведомить юзера при вводе неправильного пароля <2> система должна ограничить количество попыток ввода

Пользовательские и функциональные требования			
			<p>пароля до 5 раз, после чего временно блокировать возможность ввода на 5 минут</p> <p><3> система обязана предоставить возможность восстановления пароля через номер телефона, привязанный к аккаунту</p> <p><4> система должна уведомлять пользователя об успешной авторизации и перенаправлять его в личный кабинет</p> <p><5> система может предложить опцию "Запомнить устройство" для упрощения последующих авторизаций</p>
Client	Просмотр кешбека	Как клиент системы, я хочу просмотреть все имеющиеся кэшбэк предложения для меня и успешно выбрать	<1> система обязана отображать клиенту список всех доступных кэшбэк-предложений, включая

Пользовательские и функциональные требования

		понравившиеся	<p>персонализированные и партнерские категории</p> <p><2> система должна позволять клиенту фильтровать и сортировать предложения по категориям, размерам вознаграждений и другим параметрам</p> <p><3> система обязана предоставлять клиенту подробное описание каждой категории кэшбэка, включая условия начисления и ограничения</p>
Client	Запрос объяснений по кэшбэк	Как клиент системы, я хочу запросить разъяснение по поводу ошибок начисления и работы кэшбэка, как результат со мной свяжется администратор и решит мои вопросы	<p><1> система должна предоставлять клиенту возможность отправить запрос на разъяснение по ошибкам начисления или работе кэшбэка</p> <p><2> система обязана уведомлять администратора</p>

Пользовательские и функциональные требования			
			<p>о запросе клиента и передавать ему все необходимые данные для решения проблемы</p> <p><3> система должна обеспечивать обратную связь с клиентом, включая уведомления о статусе запроса и его решении.</p>
Client	выбор категорий кешбека	<p>Как клиент системы, я хочу выбрать нужный мне кэшбэк и, как следствие, я смогу им активно пользоваться в повседневной жизни</p>	<p><1> система должна позволять клиенту выбирать понравившиеся категории кэшбэка из списка доступных предложений</p> <p><2> система обязана сохранять выбранные клиентом категории и активировать их для начисления кэшбэка</p> <p><3> система должна уведомлять клиента об успешном</p>

Пользовательские и функциональные требования			
			выборе категорий и их активации
Client	просмотр накоплений	Как клиент системы, я хочу посмотреть свои накопления по кэшбэк и, как следствие, увидеть страницу со всеми начислениями и результат на текущий момент	<p><1> система обязана отображать клиенту страницу с текущим балансом кэшбэка и историей начислений</p> <p><2> система должна обновлять информацию о накоплениях в реальном времени после каждой транзакции</p>
Administrator	Индивидуальный запрос к партнеру по поводу уточнения оплаты	Как администратор банка, я хочу в сомнительных операциях сделать прямой запрос к партнёрам, чтобы убедиться в том, что пользователь не будет обманут и сохранить имидж компании	<p><1> система должна предоставлять администратору возможность создать индивидуальный запрос к партнеру для уточнения деталей оплаты в сомнительных операциях</p> <p><2> система обязана автоматически собирать и прикреплять к запросу все</p>

Пользовательские и функциональные требования

необходимые
данные:
информацию о
клиенте, детали
транзакции и
выбранную
категорию
кэшбэка

<3> система
должна
отправлять
запрос партнеру
через API

<4> система
обязана
уведомлять
администратора
о получении
ответа от
партнера и
предоставлять
возможность
просмотра этого
ответа в
интерфейсе

<5> система
должна
автоматически
обновлять статус
операции на
основе ответа
партнера

<6> система
обязана
уведомлять
клиента о
результатах
проверки, если
это необходимо,
и вносить
корректировки в

Пользовательские и функциональные требования			
			<p>начисления кэшбэка</p> <p><7> система должна вести лог всех запросов и ответов для дальнейшего аудита и анализа.</p>
Administrator	Редактирование неучтенных операций пользователей	<p>Как администратор банка, я хочу редактировать операции пользователей, если система не учла какие-то изменения в кэшбэке, чтобы избежать финансовых расчетов и недочетов и сохранить имидж банка</p>	<p><1> система должна предоставлять администратору доступ к списку операций пользователей, которые не были учтены или обработаны системой корректно</p> <p><2> система обязана отображать детали каждой операции: сумму, категорию кэшбэка, дату и время, статус начисления</p> <p><3> система должна позволять администратору вручную редактировать данные операции, включая изменение суммы, категории</p>

Пользовательские и функциональные требования

кэшбэка или
статуса
начисления

<4> система
обязана
автоматически
пересчитывать
кэшбэк-начислен
ия после
внесения
изменений в
операцию

<5> система
должна
уведомлять
администратора
о потенциальных
ошибках или
конфликтах при
редактировании
операции

<6> система
обязана
сохранять
историю всех
изменений,
внесенных
администраторо
м, с указанием
даты, времени и
причины
редактирования

<7> система
должна
уведомлять
пользователя о
внесенных
изменениях в его
операции, если
это необходимо

Пользовательские и функциональные требования			
			<p><8> система обязана предоставлять администратору возможность отката изменений, если редактирование было выполнено ошибочно.</p>
AI Agent	Формирование категорий кешбека	<p>Как AI агент, я хочу иметь доступ к всем категориям кешбека, данным пользователей и их тратам для того, чтобы успешно определять наиболее эффективные категории кэшбэка для пользователя и удовлетворять партнёрские программы</p>	<p><1> система должна предоставлять AI агенту доступ ко всем данным о категориях кэшбэка, включая партнерские и персонализированные предложения</p> <p><2> система обязана передавать AI агенту данные о пользователях: их демографические данные, историю транзакций, предпочтения и структуру расходов</p> <p><3> система должна позволять AI агенту анализировать данные для определения</p>

Пользовательские и функциональные требования

наиболее релевантных категорий кэшбэка для каждого пользователя

<4> система обязана учитывать при формировании рекомендаций как интересы пользователей, так и маркетинговые цели партнерских программ

<5> система должна предоставлять AI агенту возможность ранжировать категории кэшбэка по их вероятной эффективности для каждого пользователя

<6> система обязана обновлять рекомендации AI агента в реальном времени на основе новых данных о транзакциях и изменениях в предпочтениях

Пользовательские и функциональные требования			
			<p>пользователей</p> <p><7> система должна позволять AI агенту тестировать новые алгоритмы и модели для повышения точности рекомендаций</p> <p><8> система обязана предоставлять отчеты о эффективности рекомендаций AI агента, включая метрики выбора категорий пользователями и выполнение партнерских KPI</p> <p><9> система должна обеспечивать безопасность и конфиденциальность данных пользователей при их обработке AI агентом.</p>

Задание 2

users			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор пользователя. Автоматически генерируется
login	text	NOT NULL, UNIQUE	Логин пользователя для входа в систему. Должен быть уникальным
password	text	NOT NULL	Пароль пользователя. Хранится в зашифрованном виде
role_id	integer	NOT NULL, FOREIGN KEY	Идентификатор роли пользователя. Ссылается на таблицу roles
phone	varchar(11)	UNIQUE	Номер телефона пользователя. Должен быть уникальным и состоять из 11 символов
mail	text	UNIQUE	Электронная почта пользователя. Должна быть уникальной

roles			
имя атрибута	тип атрибута	ограничения	описание атрибута

roles			
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
role	varchar(63)	UNIQUE, NOT NULL	Название роли. Должно быть уникальным и не превышать 63 символов

active_cashback			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
user_id	integer	NOT NULL FOREIGN KEY	Идентификатор пользователя. Ссылается на таблицу users.
cashback_id	integer	NOT NULL FOREIGN KEY	Идентификатор активной категории кэшбэка. Ссылается на таблицу cashback.

cashback			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
name	varchar(63)	NOT NULL	Название роли. Должно быть

cashback			
			уникальным и не превышать 63 символов
description	text		Описание категории кэшбэка (необязательное поле)
is_partner	bool	NOT NULL, DEFAULT FALSE	Флаг, указывающий, является ли категория партнерской. По умолчанию FALSE
season	integer	NOT NULL	Идентификатор сезон. 0 - нет сезона 1 - осень 2- зима 3- весна 4 - лето и т.д.
start_date	timestamp	NOT NULL	Дата и время начала действия категории кэшбэка
end_date	timestamp		Дата и время окончания действия категории кэшбэка (необязательное поле).

season_cashback			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
season_name	text	NOT NULL	Название сезона.

transaction			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
user_id	text	NOT NULL, FOREIGN KEY	Идентификатор пользователя, совершившего транзакцию. Ссылается на таблицу users
payment_method	integer	NOT NULL, FOREIGN KEY	Идентификатор метода оплаты. Ссылается на таблицу payment_method
date	timestamp	NOT NULL	Дата и время совершения транзакции
status	integer	NOT NULL	Идентификатор статуса транзакции. -1 - ошибка 0 - завершена 1 - начата 2- принята 3- подтверждена

transaction_cashback			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
transaction_id	integer	NOT NULL,	Идентификатор

transaction_cashback			
		FOREIGN KEY	транзакции. Ссылается на таблицу transaction
cashback_id	integer	NOT NULL	Идентификатор кэшбэка

payment_method			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
method	text	NOT NULL	Описания метода оплаты

rewards			
имя атрибута	тип атрибута	ограничения	описание атрибута
id	integer	PRIMARY KEY, NOT NULL	Уникальный идентификатор. Автоматически генерируется
user_id	integer	NOT NULL, FOREIGN KEY	Идентификатор юзера. Ссылается на таблицу users
transaction_id	integer	NOT NULL, FOREIGN KEY	Идентификатор транзакции. Ссылается на таблицу transaction
reward_amount	decimal(10, 2)		Сумма, полученная как награда за пользование кэшбэком

rewards			
reward_date	timestamp		дата награждения

Задание 3

```
name: emptyUndead

run-name: ${github.actor}

on:
  push:
    branches: ["main"]
  pull_request:
    branches: ["main"]

jobs:
  lint_migrations:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: sbdchd/squawk-action@v1
      with:
        pattern: "docker-entrypoint-initdb.d/*.sql"
        version: "latest"

  build_docker_image:
    needs: lint_migrations
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
        run: chmod +x build.sh
      - uses: actions/checkout@v4
        run: chmod +x start.sh
```

Задание 4

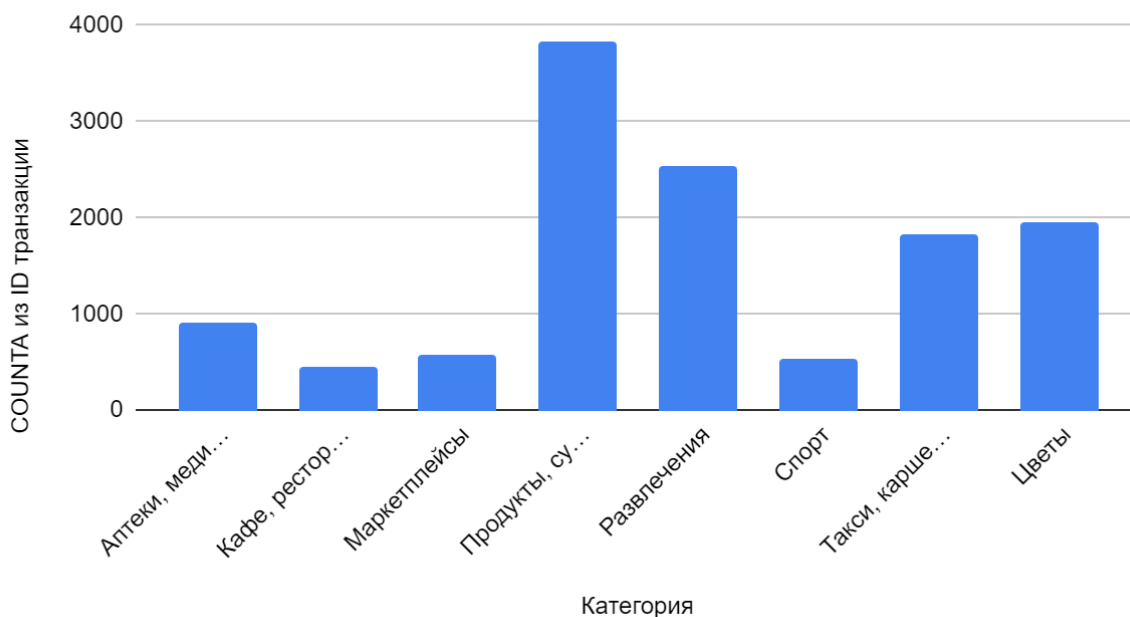
Дан csv-файл, в котором представлены исторические данные об операциях клиентов банка по категориям с указанием размера кешбэка. Предложите и обоснуйте подход по анализу представленных данных с целью выявления влияния размера кешбэка на активность покупок пользователей (в целом, по категориям товаров, по размеру кешбэка). Проведите анализ представленных в файле данных и обоснуйте выводы.

Выявления кол покупок по категориям

Шаг 1 - Построение графика распределения количества кэш бэков по категориям.

На графике видно, как распределяются кэшбэк по категориям.

COUNTA из ID транзакции



Вот распределение по категориям:

- Аптеки, медицина: 904
- Кафе, рестораны: 451
- Маркетплейсы: 576
- Продукты, супермаркеты: 3835
- Развлечения: 2535

- Спорт: 525
- Такси, каршеринг: 1823
- Цветы: 1959

Среднее количество кэшбэков по категориям составляет 1576.

Шаг 2 - Проверка на нормальное распределение.

Для проверки нормальности распределения я использую тест Шапиро-Уилика, чтобы убедиться, что количество кэшбэков по категориям имеет нормальное распределение.

```
from scipy.stats import shapiro

data = [904, 451, 576, 3835, 2535, 525, 1823, 1959]

stat, p = shapiro(data)

print(f"Статистика: {stat}, p-значение: {p}")

alpha = 0.05
if p > alpha:
    print("We do not reject H0")
else:
    print("Reject H0")
```

Так как p-значение больше 0.05, мы не отвергаем нулевую гипотезу, что данные соответствуют нормальному распределению. Это означает, что распределение кэшбэков по категориям близко к нормальному.

Шаг 3 - Статистический тест.

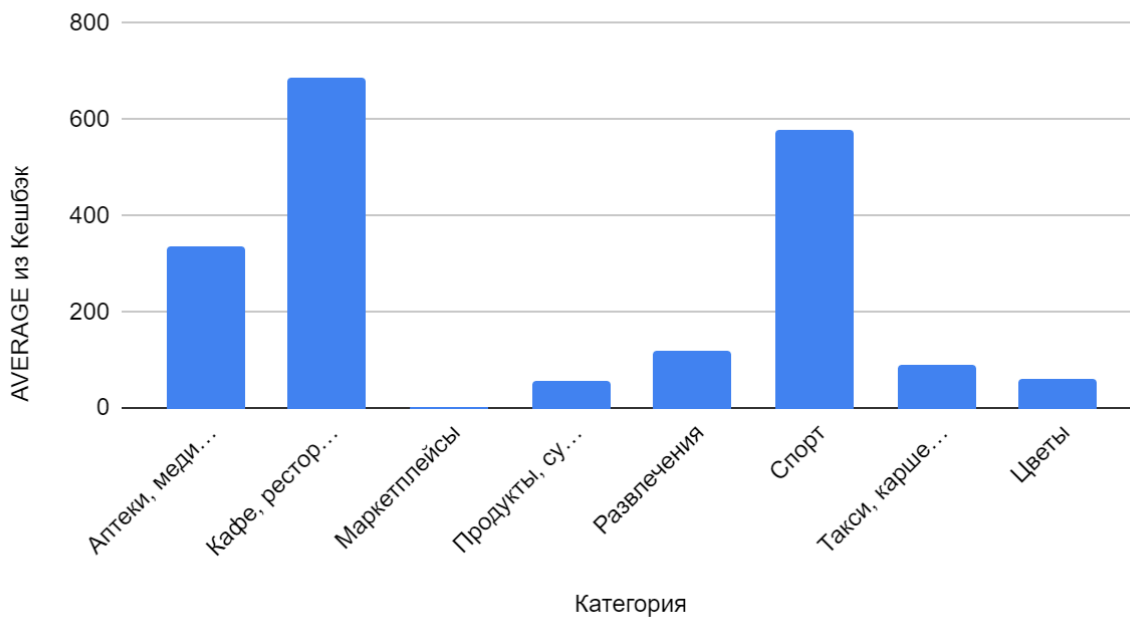
Критерий Краскела-Уоллиса применяется для сравнения медиан между группами и является устойчивым к выбросам. Полученное p-значение оказалось выше порога значимости 0.05. Это говорит о том, что нет достаточных оснований отклонить нулевую гипотезу о равенстве распределений кэшбэков между категориями. То есть, статистически значимых различий в размере кэшбэка по категориям не обнаружено.

Дисперсионный анализ показывает, что статистически значимых различий между категориями по размеру кэшбэка нет.

Выявления зависимости между кол покупок и кэшбэком.

Шаг 1 - Построение графика среднего счета кэшбэка по категориям.

AVERAGE из Кэшбэк



Вот распределение по категориям:

- Аптеки, медицина: 336
- Кафе, рестораны: 688
- Маркетплейсы: 2
- Продукты, супермаркеты: 55
- Развлечения: 121
- Спорт: 579
- Такси, каршеринг: 91
- Цветы: 60

Среднее значение кэшбэка по категориям составляет 221.375.

Шаг 2 - Проверка на нормальное распределение. Для проверки нормальности распределения я использую тест Шапиро-Уилка, чтобы удостовериться, что размер кэшбэков по категориям имеет нормальное распределение.

```
from scipy.stats import shapiro
```

```
data = [336, 688, 2, 55, 121, 579, 91, 60]

stat, p = shapiro(data)

print(f"Статистика: {stat}, p-значение: {p}")

alpha = 0.05

if p > alpha:

    print("We do not reject H0")
else:

    print("Reject H0")
```

Полученные данные:

- Статистика: 0.821457028388977
- p-значение: 0.04833751171827316

Так как p-значение меньше 0.05, мы отвергаем нулевую гипотезу и делаем вывод, что данные не соответствуют нормальному распределению.

Шаг 3 — Статистический тест.

Мы применяем критерий Краскела-Уоллиса, чтобы проверить, есть ли статистически значимые различия между медианами категорий.

Поскольку данные имеют отклонения от нормальности, критерий Краскела-Уоллиса является подходящим выбором для проверки различий.