

**Техническое задание/ Спецификация
на разработку приложения:
“Онлайн-сервис заказа пиццы вашей мечты
"PizzaCraft"**

Выпускная квалификационная работа

разработка: студент группы 110723-m-be
Франчук Светлана

преподаватель: Костяной Юрий

Содержание

1	Схема работы приложения	3
2	Описание сущностей в структуре проекта	8
3	Доменная модель	18
4	Слой персистентности	21
5	Обмен данными	23
6	Представления	27
7	Приложение 1 UserStory	28

1. Схема работы приложения

Владелец сети пиццерий принял решение о расширении спектра услуг, предоставляемых клиентам его заведений. В рамках этого решения был проведен анализ и собраны статистические данные (пользовательские истории приведены в Приложении 1), после чего было принято решение о разработке онлайн-приложения для заказа пиццы. Главной целью проекта является повышение качества предоставляемого сервиса, расширение клиентской базы и автоматизация процесса изготовления пиццы в соответствии с индивидуальными пожеланиями клиентов.

Для онлайн-приложения была предложена следующая структурная схема:

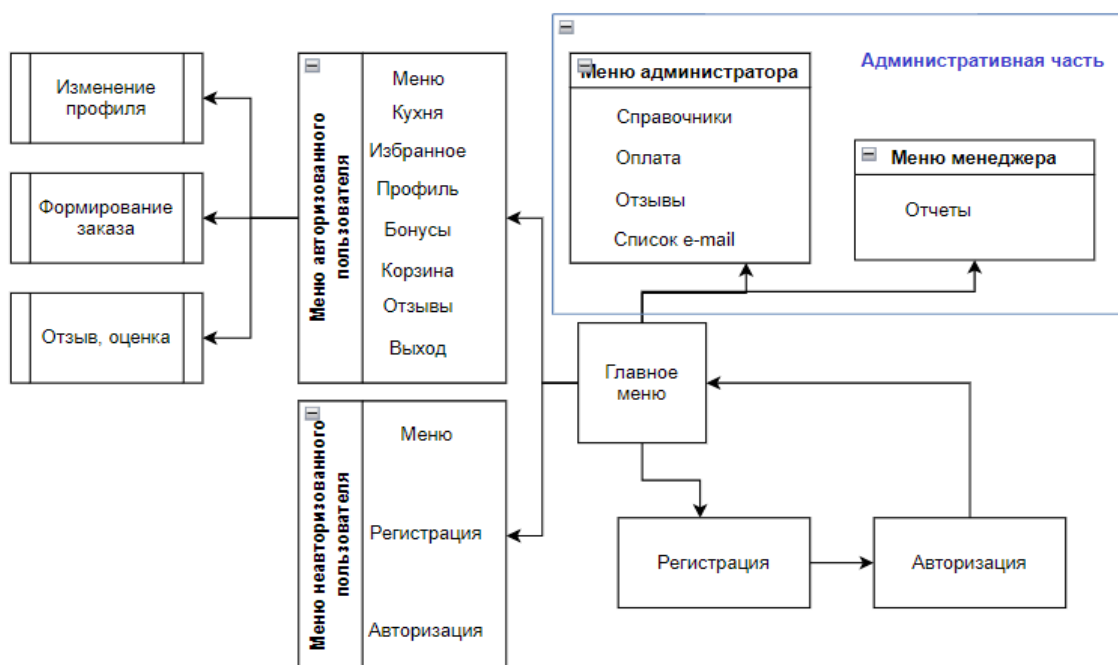
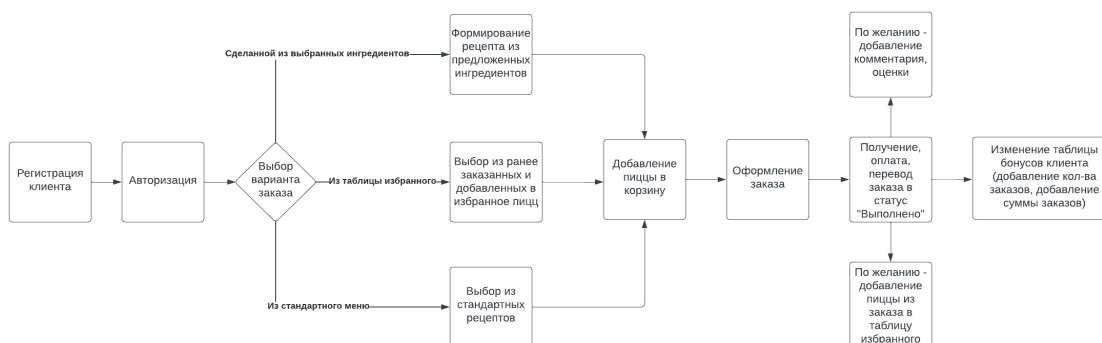


Схема работы клиента:



1) Пользователь регистрируется. При регистрации пользователь вводит следующую информацию:

- логин;

- пароль;
- E-mail - для получения в будущем информации о программах и скидках;
- дату рождения - для получения бонусных скидок при заказе в день рождения;

- адрес доставки;
- контактный телефон для связи

Дополнительно, сохраняется дата регистрации пользователя.

2) В дальнейшем для авторизации используется логин и пароль.

3) С главной страницы авторизованный пользователь имеет доступ к следующим разделам:

- меню (классические рецепты с разбивкой на виды пицц: классическая итальянская, американская, специальная (гриль или фруктовая));
- кухня (раздел, где клиент самостоятельно “готовит” пиццу, выбирая тип приготовления, тесто и ингредиенты);
- избранное (раздел в котором сохранены рецепты ранее заказанных пицц и отмеченных для сохранения в избранное);
- профиль (раздел в котором пользователь может изменить информацию о себе: адрес, телефон или пароль. Остальные поля не доступны для изменения);
- бонусы (раздел в котором видны предлагаемые бонусы, условия для их получения и накопительный текущий статус клиента. Например: для получения бесплатной пиццы нужно сделать заказ на 100 Евро. На данный момент клиент имеет историю с 57 Евро согласно предыдущих заказов);
- корзина (раздел с выбранными пиццами)
- заказы (раздел с историей заказов клиентов (содержит информацию с датой заказа, суммой и списком пицц). В списке пицц доступна возможность отметки пиццы, как избранного рецепта);
- отзывы клиентов (раздел, где пользователь может просмотреть отзывы клиентов и оценки, которые они оставили. Для пользователя есть возможность оставить отзыв и поставить свою оценку заведению)

4) Меню - страница на которой пользователь может:

- просмотреть все предлагаемые рецепты (в каждом рецепте указано: наименование, описание, калорийность и цена для минимального размера пиццы)
- отфильтровать рецепты по виду пиццы, по наполнению пиццы (мясная, рыбная, овощная, сырная, фруктовая)
- сделать заказ, выбрав понравившийся вариант, указав количество и размер пиццы (по умолчанию 1 шт и размер Small)

5) Кухня - страница на которой клиент осуществляет выбор:

- типа приготовления (классическая, глубокая, на гриле);

- типа теста для основы (сицилийское, неаполитанское, нью-йоркское, из цельнозерновой муки, из кукурузной муки);
- далее набор ингредиентов (соусы, основная начинка (мясо, рыба, колбаса), дополнительная начинка (овощи, грибы, сыры, фрукты, зелень). Наполнение ограничено следующими параметрами: максимум 3 соуса, максимум 4 вида основной начинки, максимум 7 видов дополнительной начинки). Для каждого ингредиента указано: наименование, калорийность и цена для минимального размера пиццы
- затем выбор размера пиццы и указание количества пицц (по умолчанию 1 шт и размер Small)
- клиент должен указать наименование пиццы и по желанию ввести краткое описание. По умолчанию: название (логин клиента+дата): описание - пусто.

Реализация конструктора пиццы позволит: повысить лояльность клиента к ассортименту пиццерии, при этом не потребуется человеческого труда для предоставления и сбора информации о составе пиццы, что сокращает трудозатраты и количество ошибок в заказах

6) Избранное - в данном разделе отображаются все сохраненные пользователем рецепты. Клиент может:

- посмотреть варианты пицц (в каждом рецепте указано: наименование, описание, калорийность и цена для минимального размера пиццы);
- отфильтровать рецепты по виду пиццы, по наполнению пиццы (мясная, рыбная, овощная, сырная, фруктовая);
- сделать заказ, выбрав понравившийся вариант, указав количество и размер пиццы (по умолчанию 1 шт и размер Small)

7) После выбора пицц пользователь может перейти в раздел корзина, просмотреть и при необходимости откорректировать заказ. В корзине отображаются: пицца, количество, размер, калорийность на выбранный размер пиццы, сумма на выбранный размер пиццы и общая сумма заказа. Далее можно приступить к оформлению заказа

8) Раздел оформление содержит:

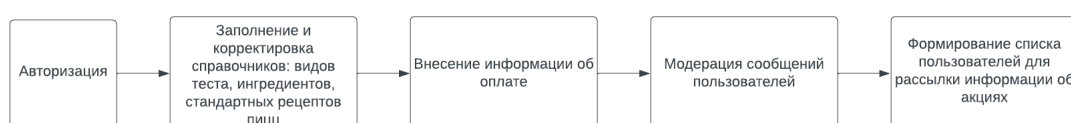
- перечень пицц;
- общую сумму заказа;
- бонус, если клиент достиг бонусного предела по сумме, количеству заказов, либо у него день рождения в день доставки, либо у заведения проходит какая-либо акция в текущий момент;
- дополнительно информацию в разделе можно дополнить временем и датой доставки. По умолчанию период доставки в пределах ближайшего часа.

Оплата производится за пределами данного приложения. В будущем приложение будет доработано интеграцией с системой оплаты . На

данном этапе администратор вносит информацию с фактом оплаты вручную, после получения подтверждения оплаты курьером.

- 9) После внесения информации об оплате в таблицу бонусов клиента добавляется информация о сумме заказа. Количество заказов клиента также увеличивается.
- 10) Неавторизованный пользователь может видеть только главную страницу с одним разделом Меню, где он может только просматривать стандартные виды пиццы (наименование, описание, калорийность, цена пиццы минимального размера) и не имеет возможность сделать заказ.

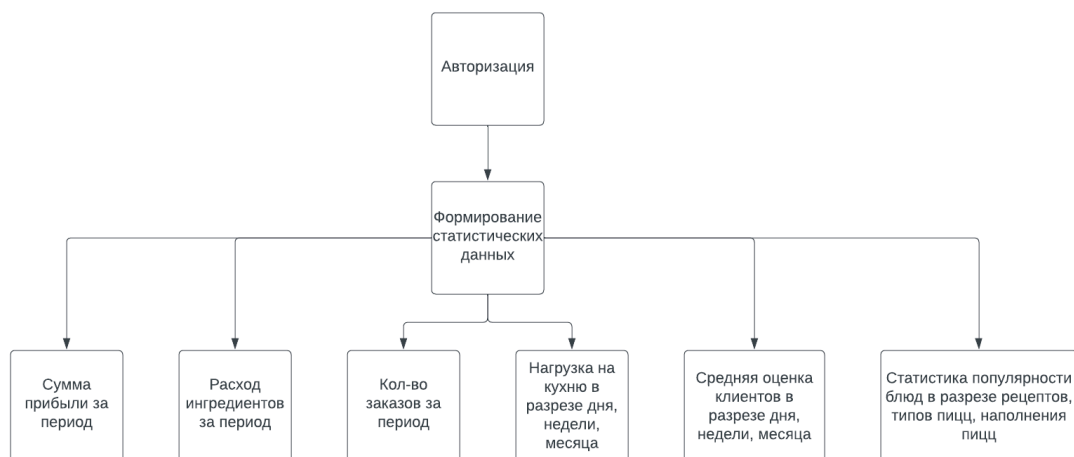
Для пользователя с ролью **администратора** предложена следующая схема:



Для администратора доступны следующие пункты меню:

- 1) Тесто - раздел в котором можно добавлять и корректировать справочник видов теста;
- 2) Ингредиенты - раздел в котором можно добавлять и корректировать справочник ингредиентов;
- 3) Пиццы - раздел в котором можно добавлять и корректировать виды пицц
- 4) Оплата - раздел в котором вносится информация об оплате заказа
- 5) Чат - раздел в котором отображаются сообщения пользователей. Здесь же есть возможность удаления и блокирования сообщений пользователя на определенный период (по умолчанию на месяц)
- 6) Статистика - раздел в котором формируется список email адресов пользователей для рассылки информации о проводимых акциях.

Для пользователя с ролью **Менеджера** доступна следующая **схема**:



Для пользователя с ролью Менеджер доступен пункт меню статистика, в котором есть возможность сформировать следующие отчеты:

- Отчет о сумме прибыли за период
- Отчет о расходе ингредиентов за период
- Отчет о количестве заказов, через приложение за период
- Отчет о нагрузке на кухню в разрезе: дня, недели, месяца
- Отчет об оценке клиентов в разрезе: дня, недели, месяца
- Отчет о статистической популярности блюд в разрезе: рецептов, типов пицц, типов наполнения пицц

2. Описание сущностей в структуре приложения

Для работы в приложении необходима первоначальная регистрация пользователя. В дальнейшем работа в приложении доступна после авторизации. Одному пользователю доступна одна роль. Для обеспечения этого функционала используется сущность **UserApp**:

№ п.п	Имя	Тип	Ограничения	Описание
1	id	Bigserial	Primary key, Not null, Unique, Auto Increment	Ключ
2	login	Varchar (20)	size (5-20), Not null, Unique	Логин
3	password	Varchar(15)	size (5-20), Not null	Пароль
4	email	Varchar (50)	size (5-50), Not null, Unique	email
5	birthdate	Timestamp	Past	Дата рождения
6	date_registration	Timestamp	Present	Дата регистрации, заполняется автоматически
7	address	Varchar (150)	size (15-150)	Адрес
8	phone_number	Varchar (15)	size (5-15)	Номер телефона
9	is_blocked	Boolean		Блокировка возможности оставлять сообщения в течение месяца с даты последнего сообщения
10	role	Enum		роль (CLIENT, MANAGER, ADMIN)
11	basket_id		Foreign key	Ссылка на таблицу Basket_id. Связь <i>один к одному</i>

Административная часть

- работы с этим функционалом доступны пользователям с ролью **ADMIN и MANAGER**;

- выполняемые **ADMIN**-ом функции:

1. Наполнение меню стандартными рецептами. В создании рецепта участвуют следующие сущности (все данные вносятся из расчета на минимальный размер пиццы. В случае выбора пользователем другого размера, все умножается на коэффициент пересчета, который находится в справочнике размеров пицц):

Ingredient

№ п.п	Имя	Тип	Ограничения	Описание
1	id	Bigserial	Primary key, Not null, Unique, Auto Increment	Ключ
2	name	Varchar (75)	size (5-20), Not null, Unique	Наименование ингредиента
3	weight	double	min(0), max(200), Not null	Вес ингредиента в пицце, грамм
4	nutrition	int	min(0), max(600), Not null	Кол-во калорий в пицце
5	price	double	min(0), Not null	Цена ингредиента в пицце
6	group	enum	Not null	Группа (SAUCE, BASIC, EXTRA)

Dough

№ п.п	Имя	Тип	Ограничения	Описание
1	id	int	Primary key, Not null, Unique, Auto Increment	Ключ
2	type_dough	Enum	not null	Типы теста

3	small_weight	double	min(0), max(200), Not null	вес на пиццу минимального размера
4	small_nutrition	int	min(0), max(600), Not null	кол-во калорий на пиццу минимального размера
5	small_price	double	min(0), Not null	цена теста для пиццы минимального размера

Pizza

№ п.п	Имя	Тип	Ограничения	Описание
1	id	BigSerial	Primary key, Not null, Unique, Auto Increment	Ключ
2	title	Varchar (25)	size (5-25), Not null, Unique	Наименование пиццы
3	description	Varchar (300)	size (0-300), Not null	Описание
4	styles	Enum	not null	Вид пиццы
5	topping_fillings	Enum	not null	Тип начинки
6	size	Enum	not null	Размер пиццы
7	ingredient_list		Foreign key	Лист ингредиентов/ Связь с таблицей ингредиентов многие ко многим
8	dough_id		Foreign key	Тесто. связь с таблицей Тесто Многие к одному
9	is_standatd_recipe	boolean	not null	признак того, что рецепт стандартный. По умолчанию - ложь

Для организации связи многие ко многим между таблицами Pizzas и Ingredients используется служебная таблица pizzas_ingredients

№ п.п	Наименование
1	ingredients_list_id
2	pizza_set_id

2. Внесение информации об оплате - изменение статуса в таблице заказы. Сущность Order будет описана ниже, в разделе функций клиента
 3. Формирование списка клиентов. В выборку должны попадать все пользователи из сущности UserApp с ролью CLIENT. Состав полей для выборки:
 - login
 - email
 4. Модерация сообщений пользователей, и блокирование пользователям возможности оставлять сообщения за нарушение правил общения на период - месяц с момента последнего общения (поле `is_Blocked` таблицы UserApp).
- выполняемые **MANAGER**-ом функции - формирование по необходимости следующих отчетов:
- 1) Отчет о сумме прибыли за период. Выборка из таблицы Order информации, ограниченной периодом дата начала и дата конца (поле `orderDateTime`) . Состав отчета: Дата, ID заказа, сумма заказа;
 - 2) Отчет о расходе ингредиентов за период. Выборка из объединения таблиц OrderDetails, Pizza, Ingredient информации, ограниченной периодом дата начала и дата конца (поле `delivery_date_time`) . Состав отчета информация сгруппированная по ID ингредиента: ID ингредиента, наименование ингредиента, суммарный вес;
 - 3) Отчет о количестве заказов, через приложение за период. Выборка из таблицы Order информации, ограниченной периодом дата начала и дата конца (поле `orderDateTime`) . Состав отчета: количество ID заказов за заданный период;
 - 4) Отчет о нагрузке на кухню в заданном периоде. Выборка из объединения таблиц OrderDetails и Pizza информации, ограниченной периодом дата начала и дата конца (поле `delivery_date_time`). Состав отчета информация сгруппированная по Дате доставки: ID пиццы, наименование пиццы, размер, количество;

- 5) Отчет об оценке клиентов в заданном периоде. Выборка из таблицы Review информации, ограниченной периодом дата начала и дата конца (поле `review_date`). Состав отчета: средняя оценка;
- 6) Отчет о статистической популярности блюд в заданном периоде в разрезе: рецептов, типов пицц, типов наполнения пицц. Выборка из объединения таблиц OrderDetails и Pizza информации, ограниченной периодом дата начала и дата конца (поле `delivery_date_time`). Состав отчета информация сгруппированная по ID пиццы: ID пиццы, наименование пиццы, тип пиццы, тип наполнения пиццы, количество заказов.

Клиентская часть

- 1) Страница Меню. Пользователю отображается выборка из таблицы Pizza всех рецептов с признаком `is_standard_recipe = true`
Просматривая страницу Пользователь может выбрать пиццу, и ее размер. Информация попадает в корзину. Содержимое корзины отражает сущность **Basket**

№ п.п	Имя	Тип	Ограничения	Описание
1	id	bigserial	Primary key, Not null, Unique, Auto Increment	Ключ
2	id_user	bigserial	Foreign key	Ссылка на таблицу UserApp. Связь Один к одному.
3	pizzas	ArrayList	Foreign key	Лист пицц. Односторонняя связь Один ко многим

Для организации связи один ко многим между таблицами Pizzas и Basket используется служебная таблица baskets_pizzas

№ п.п	Наименование
1	basket_id
2	pizzas_id

- 2) Страница Кухня. Раздел, в котором пользователь последовательно выбирает:
 - тип приготовления Enum Styles;

- типа теста для основы Enum TypeDough;
- ингредиенты из таблицы Ingredients, сгруппированные по полю `group_ingredient` (соусы, основная начинка, дополнительная начинка. Наполнение ограничено следующими параметрами: максимум 3 соуса, максимум 4 вида основной начинки, максимум 7 видов дополнительной начинки). Для каждого ингредиента указано: наименование, калорийность и цена для минимального размера пиццы
- затем выбор размера пиццы Enum TypeBySize
- в завершении клиент должен указать наименование пиццы и по желанию ввести краткое описание. По умолчанию: название (логин клиента+дата): описание - пусто.

По завершению формирования рецепта, пицца попадает в корзину

- 3) Страница Избранное. Если пользователь отметил ранее какие-либо рецепты, как понравившиеся, то на данной странице будут отражены данные из таблицы **Favorites**

№ п.п	Имя	Тип	Ограничения	Описание
1	id	bigserial	Primary key, Not null, Unique, Auto Increment	Ключ
2	id_user	bigserial	Foreign key	Ссылка на таблицу UserApp. Связь Многие к одному.
3	pizzas	ArrayList	Foreign key	Лист пицц. Односторонняя связь Многие ко многим

Для организации связи многие ко одному между таблицами Users и Favorites используется служебная таблица `users_favorites`

№ п.п	Наименование
1	<code>user_app_id</code>
2	<code>favorites_id</code>

Для организации связи многие ко многим между таблицами Pizzas и Favorites используется служебная таблица `pizzas_favorites`

№ п.п	Наименование
-------	--------------

1	pizzas_id
2	favorites_id

Пользователь может выбрать размер пиццы и количество пицц. После чего заказ попадает в корзину.

4) Страница Корзина. На данной странице отображаются:

- все пиццы, которые клиент отметил, как те, которые он хочет заказать;
- количество пицц, размер, калорийность и сумма.

После проверки заказа пользователь может перейти к оформлению заказа. Если к моменту оформления заказа пользователь накопил достаточное количество бонусов, то во время оформления заказа происходит использование бонуса. Информация о бонусах хранится в таблице **Bonuses**. Данная таблица является встраиваемой в таблицу UserApp

№ п.п	Имя	Тип	Ограничения	Описание
1	count_orders	int	min(0)	Количество заказов. При использовании бонуса, связанного с этим полем, количество обнуляется
2	sum_orders	double	min(0)	Сумма заказанных пицц. При использовании бонуса, связанного с этим полем, сумма обнуляется

Детали заказа хранят сущности **Order_details** и **Order**
Order_details

№ п.п	Имя	Тип	Ограничения	Описание
1	id	bigserial	Primary key, Not null, Unique, Auto Increment	Ключ
2	order_id	bigserial	Foreign key	Ссылка на таблицу Order. Связь Один к одному.

3	pizzas	ArrayList	Foreign key	Лист пицц. Односторонняя связь Один ко многим
4	type_bonuses	enum		тип бонуса
5	delivery_date_time	Timestamp	Future	Дата и время доставки

Для организации связи многие ко многим между таблицами Order_details и Pizzas используется служебная таблица order_details_pizzas

№ п.п	Наименование
1	order_details_id
2	pizzas_id

Orders

№ п.п	Имя	Тип	Ограничения	Описание
1	id	bigserial	Primary key, Not null, Unique, Auto Increment	Ключ
2	order_date_time	Timestamp	Present	Время формирования заказа
3	sum	double	min(0)	Общая сумма заказа
4	status_order	Enum	not null	Статус заказа, при необходимости может изменен администратором
5	order_details_id			Ссылка на таблицу Order_details. Связь Один к одному.
6	user_app_ip			Связь с таблицей UserApp - многие к одному

Для организации связи многие ко одному между таблицами Users и orders используется служебная таблица users_orders

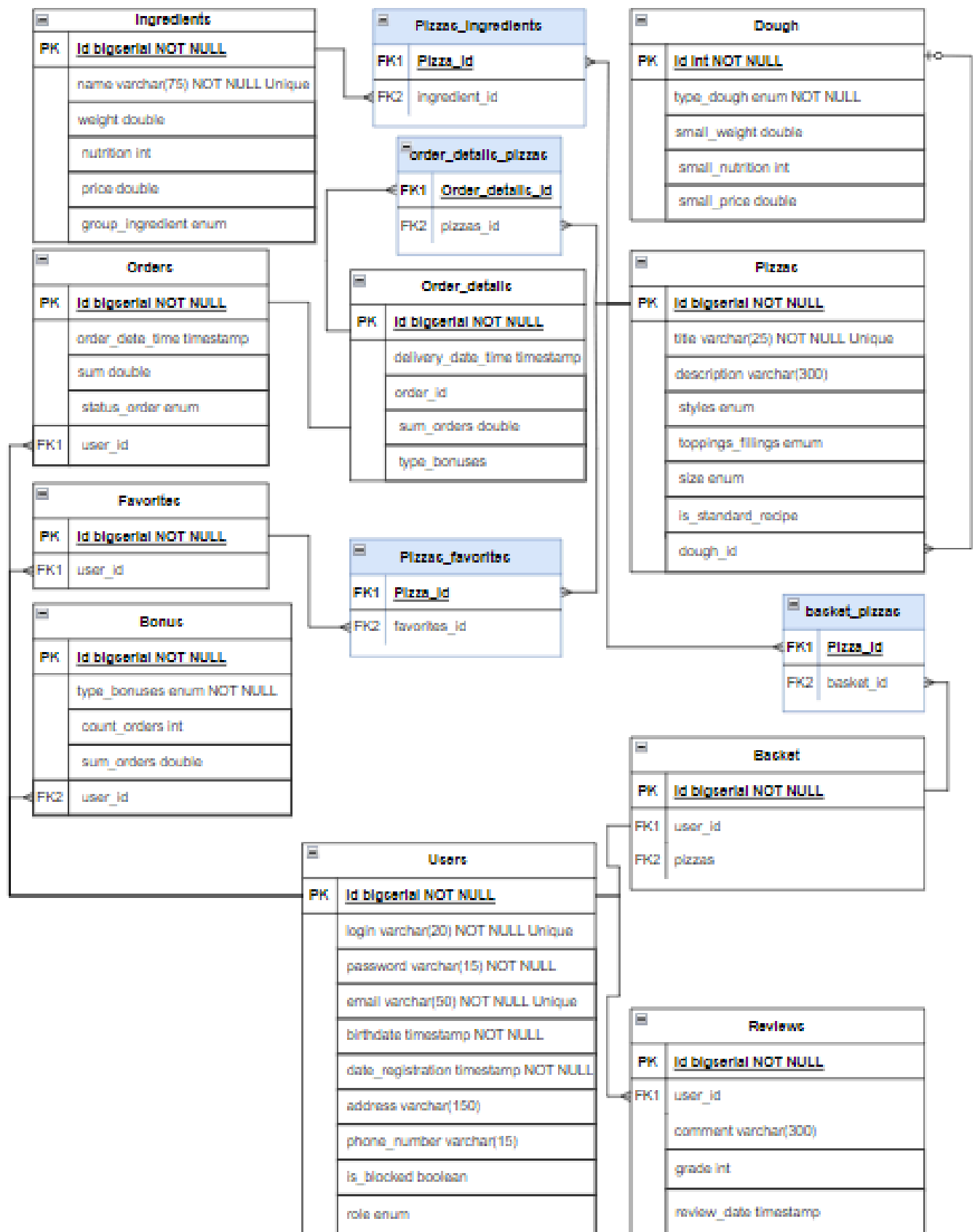
№ п.п	Наименование
1	user_app_id
2	orders_id

5) Страница Профиль - на этой странице пользователь может изменить свои данные: адрес, телефон или пароль. Изменения происходят в таблице UserApp

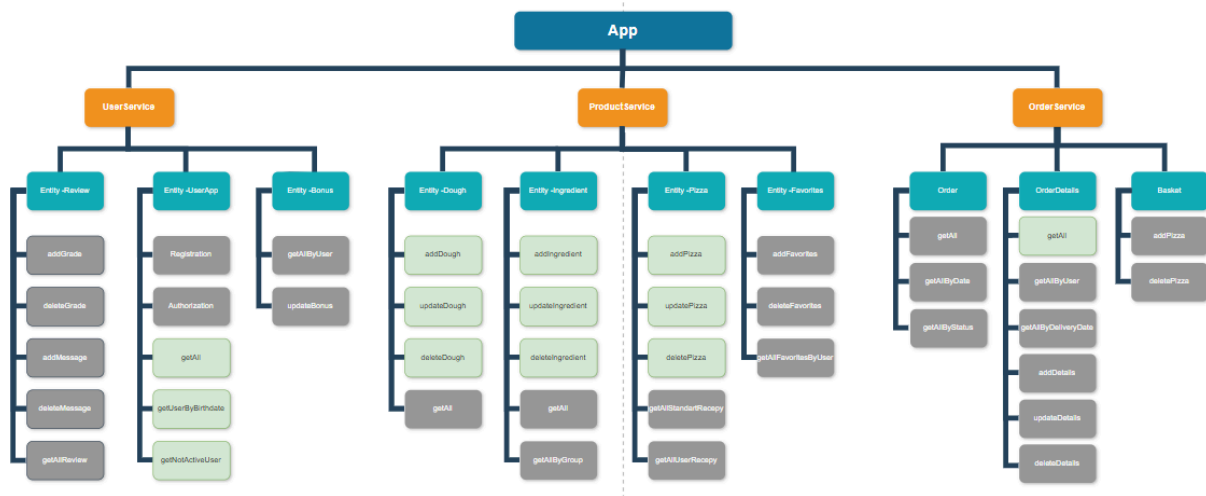
6) Страница Отзывы. Информация об отзывах и оценках пользователей содержится в таблице **Reviews**

№ п.п	Имя	Тип	Ограничения	Описание
1	id	bigserial	Primary key, Not null, Unique, Auto Increment	Ключ
2	comment	varchar(300)		Комментарий
3	grade	int	min(0)	оценка
4	review_date	Timestamp	not null	Дата внесения информации в таблицу
5	user_app_ip			Связь с таблицей UserApp - многие к одному

Диаграмма связи сущностей проекта приведена на схеме ниже.



3. Доменная модель



Доменная модель состоит из следующих сущностей, структура которых описана в предыдущем разделе:

- **UserApp** - информация о пользователях (логин, пароль, электронный адрес, адрес, телефон, дата рождения);
- **Role** - enum: ADMIN, MANAGER, CLIENT;
- **Bonus** - накопительная информация о бонусах клиента. Доступны следующие опции 50% скидка в день дня рождения, 30% скидка при заказе 30 пиццы, 100% скидка при накопленной сумме заказа пиццы 100 Евро (количество заказов, сумма заказов);
- **Review** - информация об оценках и отзывах клиентов;
- **Dough** - информация о видах теста, которые используются в пиццерии (тип пиццы, вес, калорийность и сумма для минимального размера пиццы);
- **TypeDough** - enum: CLASSICA, PAN_PIZZA, SICILIAN, NEW_YORK_STYLE, NEAPOLITAN, WHOLE_WHEAT_FLOUR, CORNMEAL;
- **Ingredient** - информация об ингредиентах, которые используются в пиццерии (наименование, вес, калорийность, цена, группа, к которой принадлежит ингредиент: соус, основная начинка, дополнительная начинка);
- **GroupIngredient** - enum: SAUCE, BASIC, EXTRA;
- **Pizza** - информация о рецептах, по которым готовится пицца (наименование, описание, тип пиццы, тип наполнения, размер, признак стандартного рецепта);
- **Styles** - enum: CLASSIC_ITALIAN, AMERICAN, SPECIALITY;
- **ToppingsFillings** - enum: MEAT, VEGETABLES, CHEESE, SEAFOOD, FRUIT;

- **TypeBySize** - enum: SMALL(coefficient 1), MEDIUM(coefficient 1.3), LARGE(coefficient 1.7). Коэффициент используется для пересчета количества необходимого теста и ингредиентов, а также для расчета калорийности блюда и его себестоимости;
- **Favorites** - информация об избранных рецептах пользователя;
- **Basket** - информация о содержании корзины пользователя, содержит список пицц, которые пользователь предполагает заказать;
- **OrderDetails** - информация с деталями заказа (дата и время доставки, список пицц, бонус, если накоплен предел);
- **TypeBonus** - enum: DISCOUNT_30, DISCOUNT_50, DISCOUNT_100;
- **Order** - общая информация о заказе (дата заказа, общая сумма заказа, статус заказа).
- **StatusOrder** - enum: NEW, CANCELED, PAID.

Для реализации алгоритмов приложения используются следующие сервисы:

- **UserService** - через данный сервис реализуются функции:
 1. авторизации пользователя;
 2. регистрации и сохранения нового пользователя;
 3. получения информации о пользователе (профиль пользователя);
 4. изменение информации в профиле пользователя;
 5. получение списка пользователей с датой рождения в заданную дату;
 6. получение полного списка пользователей с ролью CLIENT;
 7. получение списка пользователей со статусом isBlocked и датой последнего сообщения;
 8. получение списка бонусов клиента;
 9. изменение списка бонусов клиента;
 10. добавление оценки клиента;
 11. удаление оценки клиента;
 12. добавление отзыва клиента;
 13. удаление отзыва клиента
- **ProductService** - через данный сервис реализуются функции:
 1. добавления нового рецепта теста;
 2. изменения параметров существующего рецепта теста;
 3. удаление существующего рецепта теста;
 4. получения списка всех существующих рецептов теста;
 5. добавление нового ингредиента;
 6. изменение параметров существующего ингредиента;

7. удаление существующего ингредиента;
8. получение полного списка ингредиентов;
9. получение списка ингредиентов в разрезе групп;
10. получение списка ингредиентов в разрезе рецепта пиццы;
11. добавление нового рецепта пиццы;
12. изменение рецепта пиццы;
13. удаление рецепта пиццы;
14. получение списка стандартных рецептов пиццы;
15. получение списка стандартных рецептов пицц в разрезе типов приготовления;
16. получение списка стандартных рецептов пицц в разрезе типов наполнения;
17. получение списка стандартных рецептов пицц в разрезе типов приготовления и типов наполнения;
18. добавление рецепта в таблицу избранного;
19. удаление рецепта из таблицы избранного;
20. получение списка избранных рецептов пользователя

- **OrderService** - через данный сервис реализуются функции:

1. формирования корзины пользователя - добавление пиццы в корзину;
2. удаление пиццы из корзины;
3. добавление заказа;
4. удаление заказа;
5. добавление деталей заказа;
6. изменение деталей заказа - дата и время доставки, расчет бонуса;
7. удаление деталей заказа;
8. получение списка всех заказов;
9. получение списка всех заказов пользователя;
10. получение списка всех заказов в разрезе статусов;
11. получение списка всех заказов в разрезе дат.

4. Слой персистентности

Для реализации доменной модели используются следующие репозитории, связанные с таблицами в базе данных (для реализации приложения используется реляционная база данных):

- **UserAppRepository** функции:
 1. CREATE - сохранить пользователя после регистрации;
 2. READ - вернуть пользователя по логину, вернуть пользователя по email, вернуть список пользователей по дате рождения, вернуть список заблокированных пользователей, проверить существование пользователя по логину, проверить существование пользователя по email, возврат информации о доступном для расчета бонусов количестве заказов и суммы заказов;
 3. UPDATE - изменить реквизиты из профиля пользователя (адрес, телефон, пароль, состояние заблокированности), обновление информации о бонусах клиента
- **DoughRepository** функции:
 1. CREATE - создание нового рецепта теста;
 2. READ - чтение списка доступных видов теста;
 3. UPDATE - изменение реквизитов одного из видов теста;
 4. DELETE - удаление рецепта теста
- **IngredientRepository** функции:
 1. CREATE - создать новую запись об ингредиенте;
 2. READ - чтение списка ингредиентов по группе принадлежности, поиск ингредиента по имени;
 3. UPDATE - обновление реквизитов записи ингредиента;
 4. DELETE - удаление информации об ингредиенте
- **PizzaRepository** функции:
 1. CREATE - создать новый рецепт пиццы;
 2. READ - найти пиццу по имени, вернуть список рецептов пицц использующих стандартный рецепт, вернуть список пицц в разрезе типов приготовления; вернуть список пицц в разрезе типов наполнения, вернуть список пицц в разрезе типов приготовления и типов наполнения;
 3. UPDATE - обновить рецепт пиццы;
 4. DELETE - удалить рецепт пиццы
- **BasketRepository** функции:

1. CREATE - создание корзины пользователя - Создается автоматически при регистрации пользователя;
 2. READ - чтение содержания корзины по ИД пользователя;
 3. UPDATE - изменение листа пицц в корзине (добавление, удаление)
- **OrderRepository** функции:
 1. CREATE - создание заказа;
 2. READ - возврат списка всех заказов пользователя, возврат списка заказов по статусу, возврат списка заказов за указанный период дат, возврат информации по заказу пользователя;
 3. UPDATE - обновление информации о заказе
 - **OrderDetailsRepository** функции:
 1. CREATE - создание записи о деталях заказа;
 2. READ - возврат информации о деталях заказа по заказу, возврат списка заказов за указанный период доставки;
 3. UPDATE - обновление информации о деталях заказа
 - **FavoritesRepository** функции:
 1. CREATE - создание таблицы избранного для пользователя - создается в момент регистрации пользователя;
 2. READ - возврат списка избранного для пользователя;
 3. UPDATE - добавление рецепта пиццы в список избранного, удаление рецепта пиццы из списка избранного;
 - **ReviewRepository** функции:
 1. CREATE - добавление информации с оценкой и/или отзывом пользователя;
 2. READ - получение информации об оценках и отзывах пользователей
 3. DELETE - удаление информации об оценках и отзывах пользователей

5. Обмен данными

Для реализации данной функциональности необходимо создать следующие DTO:

1. UserRegistrationDTO:

- String login (логин пользователя) -
- String password (пароль пользователя)
- String email (e-mail пользователя)
- Date birthday (дата рождения пользователя)
- String deliveryAddress (адрес доставки)
- String phoneNumber (контактный телефон)

2. UserLoginDTO:

- String login (логин пользователя)
- String password (пароль пользователя)

3. UserProfileDTO:

- String login (логин пользователя)
- String deliveryAddress (адрес доставки)
- String phoneNumber (контактный телефон)
- String password (пароль пользователя)

4. DoughDTO (для администратора):

- Enum typeDough(тип теста)
- int weight (вес теста)
- int nutrition (калорийность теста)
- int price (цена теста)

5. IngredientDTO (для администратора):

- String name(наименование ингредиента)
- int weight (вес ингредиента)
- int nutrition (калорийность)
- int price (цена)
- groupIngredient (группа ингредиентов)

6. PizzaDTORequest:

- String title(наименование пиццы)
- String description (описание пиццы)
- Enum styles (тип приготовления)
- Enum toppingsFillings (тип наполнения)
- Enum size (размер)
- String Dough (тесто)

- List<IngredientSauce> (перечень соусов)
- List<IngredientBasic> (перечень основной начинки)
- List<IngredientExtra> (перечень дополнительной начинки)

7. PizzaDTOResponse:

- String title(наименование пиццы)
- String description (описание пиццы)
- int nutrition (калорийность)
- int price (цена)

8. BasketDTO:

- List<PizzaDTOResponse>

9. OrderDTO:

- List<PizzaDTOResponse> pizzas (список выбранных пицц)
- double sum (общая сумма заказа)
- Enum typeBonus (тип бонуса)
- double bonusAmount (сумма бонуса)
- Date deliveryTime (дата и время доставки)

10. ReviewDTO:

- String review (отзыв пользователя)
- int grade (оценка пользователя)

11. PizzaDTOFavorite:

- String title(наименование пиццы)
- String description (описание пиццы)
- int nutrition (калорийность)

Также необходимо создать соответствующие контроллеры для обработки запросов от пользователя:

1. UserController:

- registerUser(UserRegistrationDTO userRegistrationDTO) - метод для регистрации нового пользователя
- loginUser(UserLoginDTO userLoginDTO) - метод для авторизации пользователя
- getUserProfile(String login, UserProfileDTO userProfileDTO) - метод для получения информации о пользователе
- updateUserProfile(String login, UserProfileDTO userProfileDTO) - метод для изменения информации о пользователе
- getUserByBirthdate(LocalDate date) - метод для получения списка пользователей с датой рождения в заданную дату;

- getAllUsers () - метод для получения полного списка пользователей с ролью CLIENT
- getUserBlocked() - метод для получения списка пользователей со статусом isBlocked и датой последнего сообщения
- getUserBonus(UserProfileDTO userProfileDTO) - метод для получения списка бонусов клиента
- updateUserBonus(UserProfileDTO userProfileDTO) - метод для изменения списка бонусов клиента;
- addReviews(UserProfileDTO userProfileDTO, ReviewDTO reviewDTO) - метод для добавления оценки/отзыва клиента;
- deleteReviews(UserProfileDTO userProfileDTO, ReviewDTO reviewDTO) - метод для удаления оценки/отзыва клиента;
- getReviews() - метод для получения списка оценок/отзывов клиента;

2. ProductController:

- addRecipeDough (DoughDTO doughDTO) - метод для добавления нового рецепта теста;
- updateRecipeDough (DoughDTO doughDTO) - метод для изменения параметров существующего рецепта теста;
- deleteRecipeDough (DoughDTO doughDTO) - метод для удаления существующего рецепта теста;
- getAllDough() - метод для получения списка всех существующих рецептов теста;
- addIngredient (IngredientDTO ingredientDTO) - метод для добавление нового ингредиента;
- updateIngredient (IngredientDTO ingredientDTO) - метод для изменения параметров существующего ингредиента;
- deleteIngredient (IngredientDTO ingredientDTO) - метод для удаления существующего ингредиента;
- getAllIngredient() - метод для получения полного списка ингредиентов;
- getIngredientByGroup (IngredientDTO ingredientDTO, GroupIngredient group) - метод для получения списка ингредиентов в разрезе групп;
- getIngredientByPizza (IngredientDTO ingredientDTO, long idPizza) - метод для получения списка ингредиентов в разрезе рецепта пиццы;
- addRecipePizza (PizzaDTORequest pizzaDTORequest) - метод для добавления нового рецепта пиццы;
- updateRecipePizza (PizzaDTORequest pizzaDTORequest) - метод для изменения рецепта пиццы;
- deleteRecipePizza (PizzaDTORequest pizzaDTORequest) - метод для удаления рецепта пиццы;
- getPizzaStandardRecipe (PizzaDTOResponse pizzaDTOResponse) - метод для получения списка стандартных рецептов пиццы;

- getPizzaStandardRecepyByStyle (PizzaDTOResponse pizzaDTOResponse, Style style) - метод для получения списка стандартных рецептов пицц в разрезе типов приготовления;
- getPizzaStandardRecepyByTopping (PizzaDTOResponse pizzaDTOResponse, ToppingFilling toppingFilling) - метод для получения списка стандартных рецептов пицц в разрезе типов наполнения;
- getPizzaStandardRecepyByStyleAndTopping (PizzaDTOResponse pizzaDTOResponse, Style style, ToppingFilling toppingFilling) - метод для получения списка стандартных рецептов пицц в разрезе типов приготовления и типов наполнения;
- addFavorites (PizzaDTOFavorite pizzaDTOFavorite)- метод для добавления рецепта в таблицу избранного;
- deleteFavorites (PizzaDTOFavorite pizzaDTOFavorite) - метод для удаления рецепта из таблицы избранного;
- getFavorites (PizzaDTOFavorite pizzaDTOFavorite) - метод для получения списка избранных рецептов пользователя

3. OrderController

- addPizzaToBasket() формирования корзины пользователя - метод для добавления пиццы в корзину;
- deletePizzaFromBasket() - метод для удаления пиццы из корзины;
- addOrder() - метод для добавления заказа;
- deleteOrder - метод для удаления заказа;
- addOrderDetails - метод для добавления деталей заказа;
- updateOrderDetails - метод для изменения деталей заказа - дата и время доставки, расчет бонуса;
- deleteOrderDetails - метод для удаления деталей заказа;
- getAllOrders - метод для получения списка всех заказов;
- getUserOrders - метод для получения списка всех заказов пользователя;
- getOrdersStatus - метод для получения списка всех заказов в разрезе статусов;
- getOrdersBetweenDate - метод для получения списка всех заказов в разрезе дат.

6. Представления

В рамках реализации данного проекта предполагается разработка части бэкэнда. На следующей стадии развития проекта будет вестись разработка части фронтенда.

1. Я как пользователь, хотел бы иметь возможность регистрации в приложении, чтобы начать его использовать;
 2. Я как пользователь, хотел бы иметь доступ к моему кабинету для анализа статистики (когда были заказы, что было заказано);
 3. Я как пользователь, хотел бы иметь возможность добавления понравившегося блюда в избранное, для быстрого заказа в следующий раз;
 4. Я как пользователь, хочу иметь возможность восстановления пароля через электронную почту, для уверенности, что я не потеряю доступ к моему кабинету;
 5. Я как пользователь, хотел бы иметь возможность добавлять заказ в корзину, чтобы сформировать заказ;
 6. Я как пользователь, хотел бы иметь возможность выбора пиццы со своим набором ингредиентов, чтобы сформировать заказ в соответствии с моими пожеланиями;
 7. Я как пользователь, хотел бы иметь информацию о составе продуктов в пицце и их калорийности, для формирования заказа, отвечающего моим потребностям в заботе о своем здоровье;
 8. Я как пользователь, хотел бы иметь возможность заказать пиццу, ограниченную моим бюджетом, чтобы всегда иметь возможность заказать еду;
 9. Я как пользователь, хотел бы иметь возможность заказать не только классическую пиццу, но и другие варианты, для разнообразия моих заказов;
 10. Я как пользователь, который заботится о своем здоровье, хотел бы иметь возможность заказать вегетарианскую пиццу;
 11. Я как пользователь, хотел бы пользоваться гибкой системой бонусов, для того чтобы хотелось пользоваться только этим приложением;
 12. Я как пользователь, хотел бы иметь возможность оставлять отзывы о заведении, качестве предоставляемых услуг, вариантах заказанных блюд, а также читать отзывы остальных клиентов, для совершения быстрого выбора;
 13. Мне как пользователю, было бы приятно получать поздравления с праздниками от заведения, для формирования положительного впечатления о заведении;
 14. Мне как пользователю, хотелось бы получать рассылку от заведения о проводимых акциях и скидках, для осуществления заказов по выгодным ценам
15. Я как администратор системы, хотел бы иметь возможность простого добавления информации об ингредиентах, их калорийности и стоимости, для экономии моего рабочего времени;
 16. Я как администратор системы, хотел бы иметь возможность легкого поиска и корректировки информации о пиццах, их составах, ингредиентах, для экономии моего рабочего времени и минимизации ошибок;
 17. Я как администратор системы, хотел бы иметь возможность легкой фильтрации информации о пользователях для предоставления им обратной связи (поздравления, рассылка информации об акциях), для выполнения моих обязанностей;

18. Я как администратор системы, хотел бы иметь статистику активности пользователей для стимулирования их активности, для повышения прибыли заведения;
19. Я как администратор системы, хотел бы иметь возможность блокирования пользователю доступа к возможности оставлять комментарии за нарушение правил, чтобы поддерживать порядок в системе
20. Я как владелец заведения, хотел бы иметь статистику в разрезе периодов нагрузки на заведение, для планирования загрузки поваров;
21. Я как владелец заведения, хотел бы иметь статистику в разрезе популярности блюд, используемых ингредиентов, для планирования закупаемых продуктов;
22. Я как владелец заведения, хотел бы иметь информацию о суммах прибыли за определенные периоды, для планирования расширения услуг;
23. Я как владелец заведения, хотел бы иметь возможность анализа отзывов о заведении, для улучшения качества обслуживания