

FeedEm проект для предзаказа еды.

FeedEm используется для предзаказов еды в популярных ресторанах.

Роли пользователей:

- Покупатель
- Продавец
- Администратор

Сценарии использования

BG-1: Уменьшить время ожидания для покупателей.

F1-1: Заказ покупателя будет готов, когда он прибудет в ресторан.

Как покупателю, многим хотелось бы получать свои заказы так быстро, как это возможно.

Данное приложение позволит вам совершать предзаказ и получать еду к назначенному времени.

F1-2: Никаких очередей.

Сервис feeEm не имеет очередей ожидания. Всё, что понадобится для заказа - соединение с сетью интернет.

BG-2: Простейший путь подготовки заказа для клиентов.

F2-1: Не требуется доплата за сервис доставки.

FeedEm создан не для доставки, а для предзаказов. Цель продавцов - подготовка еды к нужному для клинтов времени. Покупатели сам придут, чтобы получить заказ.

F2-2: Не требуется создавать собственное приложение.

BurgerKing™ имеет собственное приложение для предзаказов, но написание подобного рода приложений затратно. И мы предлагаем вам наше решение.

F2-3: Бесплатная реклама.

Если предлагаемая вами еда высоко ценится клиентами, то новые покупатели сами найдут вас. Чем выше будет ваш рейтинг, тем больше людей узнают о вас.

BG-3: Легко управлять сервисом feedEm

F3-1: Получать сводную статистику

Как администратор вы хотите знать, что происходит в вашей системе.

В данном приложении вы можете просматривать различные данные в админ. панели или посмотреть общую сводку.

F3-2: Просматривать информацию о заказах.

Как администратор вы хотите просматривать все заказы в вашей системе и их статусы.

Вы можете просматривать всю эту информацию в администраторской панели.

F3-3: Добавлять или удалять продавцов.

Как администратор вы хотите добавлять или удалять продавцов.

Вы можете добавлять и удалять их в админ. панели.

Компоненты

Модели (М в модели MVC)

Смотрите ER-диаграмму.

Реализованы как модели в Ruby on rails.

Данные хранятся в PostgreSQL.

- Покупатели
- Кредитные карточки
- Продавцы
- Товары
- Заказы

Отображения (V в модели MVC)

Реализованы как отображения в Ruby on rails.

Используется Vue js фреймворк на стороне клиента.

- Домашняя страница
- Страница поиска
- Страница товара
- Страница заказа товара
- Страница профиля пользователя
- Страница входа для пользователя
- Страница регистрации пользователя
- Страница профиля продавца
- Страница входа для продавца
- Панель администратора

Контроллеры (C в модели MVC)

Реализованы как контроллеры в Ruby on Rails.

- Контроллер пользователя
- Контроллер продавца
- Контроллер заказа
- Контроллер товара
- Контроллер кредитных карт