Диаграммы Controllers, Views и Models не содержат некоторых классов, необходимость которых была выявлена в процессе разработки. Была переделана модель потому, что она не позволяла реализовать некоторый функционал необходимый для работы. Сущность Trade и Transaction были объединены в одну сущность Transaction потому, что не было необходимости в разбиении на две сущности. Models не содержали функционал, который был необходим для полной обработки выполнения всех операций. Category не позволяли реализовать подкатегории, поэтому была реализована сущность которая содержит саму себя и таким образом можно добавлять иерархию категорий. В некоторых случаях были использованы enum, когда можно было добавить новые сущности для полной гибкости бизнес процессов. Сущность Item не позволяла реализовать, что товар может иметь три состояния, когда он доступен только для обмена, покупки или же и обмена и покупки. Не было сущности для реализации системы избранных товаров. Вместо того, чтобы хранить список необходимых сущностей в самом singleton сервисах, необходимо иметь доступ к подключению к базе данных. Работа с базой данных предоставляет более гибкий подход в реализации функционала, а также она более быстро обрабатывает все необходимые запросы в отличие от процедурного стиля. В описании указано, что можно сортировать товары по количеству купленных/обмененных товаров. Не совсем было ясно к каким сущностям относится данная категория, т.к. в слое модели не были указаны такие поля, как для товара и для пользователя. По описанию системы можно только продавать по одному товару, потому что нет описания количества каждого товара. А для пользователя данную категорию не логично выделять потому, что он может продавать много недорогих товаров и быть в лидерах продаж, что не всегда отражает качество товаров. В подсистемах программы указана система уведомлений, но нигде больше не была вынесена. Я предполагаю, что тут подразумевалось направление развития программы. Не были указано свойство для хранения изображения товара, только для файла.

В использовании singleton не было необходимости, т.к в языке программирования Python нет необходимости в создании классов на каждый отдельный файл. За счёт этого можно не использовать данных подход проектирования. Это всё достигается с помощью модулей и области видимости.

Указано, что администратор может создавать новые аккаунты пользователей, но на макете главной страницы администратора такой возможности не указано. Страница создания аккаунта подрузамевает не создание новых аккаунтов пользователя, а выдачу прав администратора, так как содержится только поле для электронной почты.

В модели нет полей для отслеживания количества подтвержденных жалоб на товары каждого отедльного пользователя из-за этого нет способа реализации того, что при многократной подтверждении жалоб пользователь блокируется. Поэтому было реализованна блокировка пользователя, если хотя одна жалоба подтверждена, но она не перманентная, а временная(на один день).

Страницы просмотра товара каким-либо пользователей(администратором или обычным пользователем) не предоставляет полной информации об способах преобретения товара только можно ли обменять, когда можно еще и только купить товар.

В макетах пользовательского интерфейса не указано как отображаются история купленных или обмененных товаров. При реализации пользовательского интерфейса было предложено использование MDI форм. Данных подход, однако, не используется для реализации таких систем потому, что интерфейс должен быть удобен и понятен даже неопытному пользователю и он может легко запутаться.

Страница редактирования пользователя не позволяет редактировать все поля пользователя, описанные в текстовом описании.

В текстовом описании указано, что укзываются при регистрации электронная почта, пароль и никнейм, но в макетах интерфейсах указаны только электронная почта и пароль.

В текстовом описаниии указано, что можно в поиске искать товар не только по категориям и подкатегориям, однако на макете интерфейса указано обратное.

Присутствует кнопка назад, когда не является необходимым в данном контексте приложения, поскольку MDI формы уже предполагают наличие механизма для управления окнами и переходами между ними.