**Тестирование**

**Функциональное тестирование**

Проводилась проверка работоспособности всех ключевых функций системы в соответствии с техническим заданием для каждой роли пользователя в десктопном (WPF) и мобильном (Xamarin.Forms) приложениях. Ниже приведены примеры выполненных тест-кейсов:

**Тест-кейс Ф1**

|  |  |
| --- | --- |
| TestCase # | Ф1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Клиент: Успешная авторизация |
| Резюме испытания | Проверка возможности входа в систему для зарегистрированного клиента с корректными данными. |
| Шаги тестирования | 1. Запустить мобильное приложение. 2. На экране входа ввести корректный Email клиента. 3. Ввести корректный пароль клиента. 4. Нажать кнопку "ВОЙТИ". |
| Данные тестирования | Email: [client@gmail.com](https://www.google.com/url?sa=E&q=mailto%3Aclient%40gmail.com), Пароль: 111111. |
| Ожидаемый результат | Пользователь успешно авторизуется. Открывается главный экран приложения для клиента (с вкладками Меню, Рестораны, Корзина, Профиль). |
| Фактический результат | Соответствует |
| Предпосылки | Существует активная учетная запись клиента с указанными Email/Паролем. Приложение установлено. Доступно подключение к Интернету и Supabase. |
| Постусловия | Установлена активная сессия пользователя в приложении. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Базовый сценарий входа. |

**Тест-кейс Ф2**

|  |  |
| --- | --- |
| TestCase # | Ф2 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Клиент: Добавление и удаление из корзины |
| Резюме испытания | Проверка возможности добавления товара в корзину и его последующего удаления. |
| Шаги тестирования | 1. Авторизоваться как Клиент. 2. Перейти в "Меню". 3. Выбрать блюдо и добавить в корзину. 4. Перейти в "Корзину". 5. Убедиться, что блюдо добавлено. 6. Нажать кнопку удаления ("✕") для этого блюда. |
| Данные тестирования | Любое доступное блюдо. |
| Ожидаемый результат | 1. Блюдо успешно добавляется в корзину. 2. Блюдо успешно удаляется из корзины. 3. Итоговая сумма корзины пересчитывается корректно после добавления и удаления. |
| Фактический результат | Соответствует |
| Предпосылки | Пользователь авторизован как Клиент. В меню есть доступные блюда. |
| Постусловия | Корзина пуста (или содержит другие товары, если они были добавлены ранее). |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка основного функционала корзины. |

**Тест-кейс Ф3**

|  |  |
| --- | --- |
| Поле | Описание |
| TestCase # | Ф3 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Сотрудник: Отображение нового заказа (Realtime) |
| Резюме испытания | Проверка автоматического появления нового заказа в списке текущих заказов сотрудника без ручного обновления, используя Realtime. |
| Шаги тестирования | 1. Открыть десктопное приложение и авторизоваться как Сотрудник. 2. Перейти в раздел "Текущие заказы". 3. С помощью мобильного приложения (или другого инструмента) создать новый заказ (клиентом). 4. Наблюдать за разделом "Текущие заказы" в десктопном приложении. |
| Данные тестирования | Учетная запись сотрудника. Возможность создать новый заказ. |
| Ожидаемый результат | Новый заказ (созданный на шаге 3) автоматически появляется в списке "Текущие заказы" сотрудника в течение короткого времени (нескольких секунд) без необходимости обновлять страницу вручную. |
| Фактический результат | Соответствует |
| Предпосылки | Сотрудник авторизован и находится в разделе "Текущие заказы". Realtime-подписка для заказов настроена и активна. |
| Постусловия | Список текущих заказов актуализирован. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка ключевой функции Realtime для сотрудника. |

**Тест-кейс Ф4**

|  |  |
| --- | --- |
| TestCase # | Ф4 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Администратор: Создание категории блюд |
| Резюме испытания | Проверка возможности создания новой категории блюд администратором. |
| Шаги тестирования | 1. Авторизоваться как Администратор. 2. Перейти в раздел "Категории блюд". 3. Нажать "Добавить новую". 4. Ввести уникальное название новой категории. 5. (Опционально) Выбрать/загрузить иконку. 6. Нажать "Добавить". |
| Данные тестирования | Название: "Тестовая Категория 99", (опционально) URL или файл иконки. |
| Ожидаемый результат | 1. Новая категория появляется в списке категорий. 2. Категория сохранена в БД. 3. Иконка (если добавлена) отображается корректно. 4. Категория доступна для выбора при добавлении/редактировании блюд. |
| Фактический результат | Соответствует |
| Предпосылки | Пользователь авторизован как Администратор. |
| Постусловия | В системе существует новая категория блюд. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка CRUD операций для категорий. |

**Тест-кейс Ф5**

|  |  |
| --- | --- |
| TestCase # | Ф5 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Администратор: Попытка удаления пользователя с существующими заказами |
| Резюме испытания | Проверка реакции системы при попытке администратора удалить пользователя (клиента или сотрудника), у которого есть связанные записи в таблице заказов (что запрещено ограничением внешнего ключа). |
| Шаги тестирования | 1. Убедиться, что существует пользователь (например, Клиент), у которого есть хотя бы один заказ в таблице Заказы. 2. Авторизоваться как Администратор в десктопном приложении. 3. Перейти в раздел "Пользователи". 4. Выбрать пользователя из шага 1. 5. Нажать "Удалить выбранного". 6. Подтвердить удаление. |
| Данные тестирования | Учетная запись администратора. Учетная запись пользователя, имеющего записи в таблице Заказы. |
| Ожидаемый результат | Система должна предотвратить удаление пользователя. Должно отобразиться понятное для пользователя сообщение об ошибке, информирующее, что пользователя нельзя удалить, так как с ним связаны заказы (или другие данные). Пользователь не должен быть удален из БД. |
| Фактический результат | Приложение отображает неинформативное сообщение об ошибке, содержащее технические детали СУБД ("Update violates foreign key constraint 'Заказы\_id\_клиента\_fkey' on table 'Заказы'"). |
| Предпосылки | Администратор авторизован. Существует пользователь с записями в таблице Заказы. В БД настроено ограничение FOREIGN KEY "Заказы\_id\_клиента\_fkey" ... ON DELETE RESTRICT. |
| Постусловия | Пользователь не удален. Отображено некорректное/непонятное сообщение об ошибке. |
| Статус (Pass/Fail) | Fail |
| Комментарии | Ошибка в обработке исключений базы данных на уровне приложения. Требуется реализовать перехват конкретной ошибки СУБД (нарушение внешнего ключа при удалении) и вывод пользователю адекватного сообщения. |

**Тест-кейс Ф6**

|  |  |
| --- | --- |
| TestCase # | Ф6 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Курьер: Просмотр доступных заказов |
| Резюме испытания | Проверка корректности отображения заказов, готовых к доставке, в мобильном приложении курьера. |
| Шаги тестирования | 1. Сотрудник: Создать заказ с типом "Доставка" и перевести его в статус "Ожидает курьера". 2. Авторизоваться как Курьер в мобильном приложении. 3. Перейти на вкладку "Доступные". |
| Данные тестирования | Заказ со статусом "Ожидает курьера". Учетная запись курьера. |
| Ожидаемый результат | Заказ, созданный на шаге 1, отображается в списке доступных заказов курьера с корректным номером, адресом и суммой. |
| Фактический результат | Соответствует |
| Предпосылки | Существует заказ для доставки в статусе "Ожидает курьера". Курьер авторизован. |
| Постусловия | Список доступных заказов отображен. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка фильтрации заказов для курьера. |

**Тест-кейс Ф7**

|  |  |
| --- | --- |
| TestCase # | Ф7 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Клиент: Одновременное создание заказов |
| Резюме испытания | Проверка поведения системы при попытке одновременного создания заказов двумя клиентами с одинаковым типом (например, "Самовывоз"), что может привести к конфликту номеров из-за клиентской генерации. |
| Шаги тестирования | 1. Подготовить два устройства/эмулятора с мобильным приложением. 2. Авторизоваться под двумя разными клиентами. 3. На обоих устройствах добавить товары в корзину. 4. Одновременно на обоих устройствах нажать "Оформить заказ" и выбрать тип "Самовывоз". |
| Данные тестирования | Две учетные записи клиентов. Любые блюда. |
| Ожидаемый результат | Оба заказа должны быть успешно созданы с разными уникальными номерами заказов (например, С-12 и С-13). Не должно возникать ошибок "duplicate key value" из-за номера заказа. |
| Фактический результат | Один из клиентов получает ошибку сохранения заказа (с упоминанием "duplicate key"). |
| Предпосылки | Две учетные записи клиентов. Клиентская логика генерации номера заказа. |
| Постусловия | Как минимум один заказ не создан. |
| Статус (Pass/Fail) | Fail |
| Комментарии | Критическая ошибка при генерации номера заказа на клиенте. Необходимо перенести генерацию номера на сторону базы данных. |

**Тест-кейс Ф8**

|  |  |
| --- | --- |
| Поле | Описание |
| TestCase # | Ф8 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Администратор: Поиск пользователя по части email |
| Резюме испытания | Проверка функциональности поиска пользователя в разделе администрирования по части его email адреса. |
| Шаги тестирования | 1. Авторизоваться как Администратор. 2. Перейти в раздел "Пользователи". 3. Убедиться, что в списке есть несколько пользователей с разными email. 4. В поле поиска ввести часть email одного из пользователей (например, "@gmail"). |
| Данные тестирования | Список пользователей с различными email. Часть email для поиска. |
| Ожидаемый результат | В таблице остаются только те пользователи, чей email содержит введенную в поле поиска подстроку. |
| Фактический результат | Соответствует |
| Предпосылки | Пользователь авторизован как Администратор. Существует несколько пользователей с email, содержащими искомый фрагмент. |
| Постусловия | Список пользователей отфильтрован. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка работы текстового поиска. |

**Модульное тестирование (Unit Testing)**

Для проверки корректности работы отдельных логических компонентов системы было проведено модульное тестирование. Основное внимание уделялось проверке логики. Ниже приведены примеры выполненных тест-кейсов:

**Тест-кейс М1**

|  |  |
| --- | --- |
| TestCase # | М1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Desktop: Корректность текста кнопки для статуса "Принят" |
| Резюме испытания | Проверка, что для заказа со статусом "Принят" модуль интерфейса предлагает корректное действие ("Готов"). |
| Шаги тестирования | 1. Подготовить тестовые данные, соответствующие статусу "Принят". 2. Передать эти данные в модуль, отвечающий за текст кнопок. |
| Данные тестирования | Идентификатор статуса "Принят". |
| Ожидаемый результат | Модуль возвращает текстовую строку "Готов". |
| Фактический результат | Соответствует |
| Предпосылки | Модуль (конвертер) для определения текста кнопки существует. Идентификаторы статусов известны. |
| Постусловия | Нет. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка базовой логики отображения действий для статуса. |

**Тест-кейс М2**

|  |  |
| --- | --- |
| TestCase # | М2 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Desktop: Применение фильтра по имени пользователя |
| Резюме испытания | Проверка, что при вводе текста в поле поиска пользователей список отображаемых пользователей корректно фильтруется по имени. |
| Шаги тестирования | 1. Подготовить тестовый набор данных пользователей (с разными именами). 2. Передать этот набор в модуль управления пользователями. 3. Задать критерий поиска (часть имени). 4. Вызвать функцию применения фильтра. |
| Данные тестирования | Список пользователей: {"Иван Петров", "Петр Иванов", "Анна Сидорова"}. Критерий поиска: "Иван". |
| Ожидаемый результат | В результате работы модуля отфильтрованный список содержит только пользователей "Иван Петров" и "Петр Иванов". |
| Фактический результат | Соответствует |
| Предпосылки | Модуль управления пользователями (ViewModel) существует. Функция фильтрации реализована. |
| Постусловия | Внутреннее состояние модуля (список для отображения) изменено. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка логики фильтрации на тестовых данных. |

**Тест-кейс М3**

|  |  |
| --- | --- |
| TestCase # | М3 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Desktop: Активация возможности редактирования пользователя при выборе |
| Резюме испытания | Проверка, что модуль управления пользователями корректно определяет возможность редактирования (активирует соответствующую функцию) только при выборе пользователя из списка. |
| Шаги тестирования | 1. Имитировать состояние, когда в списке пользователей не выбрана ни одна запись. 2. Проверить внутреннее состояние модуля на предмет доступности функции редактирования. 3. Имитировать выбор одного пользователя в списке. 4. Повторно проверить состояние модуля на предмет доступности функции редактирования. |
| Данные тестирования | Два состояния: "пользователь не выбран" и "пользователь выбран". |
| Ожидаемый результат | 1. При отсутствии выбора пользователя, функция редактирования недоступна (внутренний флаг/состояние = false). 2. При выборе пользователя, функция редактирования становится доступна (внутренний флаг/состояние = true). |
| Фактический результат | Соответствует |
| Предпосылки | Существует модуль (ViewModel) управления пользователями с логикой определения доступности действий. |
| Постусловия | Внутреннее состояние модуля корректно отражает возможность редактирования в зависимости от выбора. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка логики управления доступностью действий на основе состояния выбора данных. |

**Тест-кейс М4**

|  |  |
| --- | --- |
| TestCase # | М4 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Desktop: Применение фильтра пользователей при отсутствии данных |
| Резюме испытания | Проверка поведения модуля фильтрации пользователей, если список пользователей еще не был загружен (например, ошибка сети при инициализации). Ожидается корректная обработка ситуации, но тест проверяет наличие ошибки. |
| Шаги тестирования | 1. Имитировать ситуацию, когда список пользователей не загружен (внутренний список равен null или пуст). 2. Задать критерий поиска. 3. Вызвать функцию применения фильтра. |
| Данные тестирования | Состояние "список пользователей не загружен". Критерий поиска: "test". |
| Ожидаемый результат | Модуль должен корректно обработать ситуацию отсутствия данных: не должно возникать критических ошибок (сбоев), итоговый отображаемый список должен быть пустым. |
| Фактический результат | Происходит сбой в работе модуля NullReferenceException при попытке отфильтровать несуществующий список. |
| Предпосылки | Модуль управления пользователями существует. Имитируется состояние отсутствия данных. |
| Постусловия | Ошибка выполнения модуля. |
| Статус (Pass/Fail) | Fail |
| Комментарии | Отсутствует проверка на наличие данных перед их обработкой в функции фильтрации. Требуется добавить проверку на null |

**Тест-кейс М5**

|  |  |
| --- | --- |
| TestCase # | М5 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Mobile: Расчет суммы пустой корзины |
| Резюме испытания | Проверка, что модуль корзины корректно возвращает нулевую сумму и нулевое количество товаров для пустой корзины. |
| Шаги тестирования | 1. Получить доступ к модулю управления корзиной. 2. Убедиться, что корзина пуста (или очистить ее). 3. Запросить итоговую сумму и общее количество товаров. |
| Данные тестирования | Пустая корзина. |
| Ожидаемый результат | Итоговая сумма равна 0. Общее количество товаров равно 0. |
| Фактический результат | Соответствует |
| Предпосылки | Модуль корзины (CartService) доступен. |
| Постусловия | Нет. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка базового состояния корзины. |

**Тест-кейс М6**

|  |  |
| --- | --- |
| TestCase # | М6 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Mobile: Уменьшение количества товара в корзине до нуля |
| Резюме испытания | Проверка, что при уменьшении количества товара, равного 1, этот товар удаляется из корзины. |
| Шаги тестирования | 1. Добавить в корзину один экземпляр товара А. 2. Убедиться, что товар А есть в корзине с количеством 1. 3. Вызвать функцию уменьшения количества для товара А. |
| Данные тестирования | Товар А. |
| Ожидаемый результат | Товар А удаляется из корзины. Количество товаров в корзине становится 0 (если других не было). Итоговая сумма становится 0 (если других не было). |
| Фактический результат | Соответствует |
| Предпосылки | Модуль корзины доступен. |
| Постусловия | Товар удален из корзины. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка логики удаления при уменьшении количества до 1. |

**Тест-кейс М7**

|  |  |
| --- | --- |
| TestCase # | М7 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Mobile: Активация кнопки оформления заказа при непустой корзине |
| Резюме испытания | Проверка, что кнопка "Оформить заказ" на экране корзины становится доступной для нажатия, как только в корзине появляется хотя бы один товар. |
| Шаги тестирования | 1. Убедиться, что корзина пуста. 2. Проверить состояние кнопки "Оформить заказ". 3. Добавить один любой товар в корзину. 4. Повторно проверить состояние кнопки "Оформить заказ". |
| Данные тестирования | Любой доступный товар. |
| Ожидаемый результат | 1. Изначально кнопка "Оформить заказ" неактивна (шаг 2). 2. После добавления товара кнопка "Оформить заказ" становится активной (шаг 4). |
| Фактический результат | Соответствует |
| Предпосылки | Модуль (страница) корзины существует. Логика активации/деактивации кнопки оформления заказа реализована. |
| Постусловия | Кнопка "Оформить заказ" активна. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка базовой логики UI, предотвращающей оформление пустого заказа. |

**Тест-кейс М8**

|  |  |
| --- | --- |
| TestCase # | М8 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Mobile: Оформление заказа при недоступном товаре в корзине |
| Резюме испытания | Проверка поведения системы при попытке оформить заказ, если в корзине оказался товар, который стал недоступным *после* добавления в корзину. Ожидается проверка доступности перед оформлением, но тест проверяет отсутствие этой проверки. |
| Шаги тестирования | 1. Клиент: Добавить доступный товар А в корзину. 2. Администратор: Сделать товар А недоступным. 3. Клиент: Не обновляя корзину, нажать "Оформить заказ". 4. Клиент: Выбрать любой тип заказа и подтвердить. |
| Данные тестирования | Товар А, который можно сделать недоступным. Учетные записи клиента и администратора. |
| Ожидаемый результат | Система перед созданием заказа должна проверить актуальную доступность всех товаров в корзине. При обнаружении недоступного товара должно отобразиться сообщение об ошибке с указанием, какой товар нужно убрать из корзины. Заказ не должен быть создан. |
| Фактический результат | Система не проверяет актуальную доступность товаров при оформлении заказа. Заказ успешно создается в БД, включая в свой состав недоступный товар. |
| Предпосылки | Клиент авторизован. Администратор имеет возможность менять доступность товара. Товар изначально доступен. |
| Постусловия | Заказ создан с недоступным товаром. |
| Статус (Pass/Fail) | Fail |
| Комментарии | Ошибка в логике оформления заказа. Необходимо добавить проверку доступности товаров из корзины непосредственно перед созданием записи заказа в базе данных. |

**Тестирование пользовательского интерфейса (UI/Usability)**

Проводилась оценка удобства и корректности работы пользовательского интерфейса мобильного приложения «Курочка и точка». Ниже приведены примеры выполненных тест-кейсов:

**Тест-кейс И1**

|  |  |
| --- | --- |
| TestCase # | И1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Mobile: Навигация по основным вкладкам (Клиент) |
| Резюме испытания | Проверка плавности и корректности переключения между основными экранами (вкладками) приложения для клиента. |
| Шаги тестирования | 1. Авторизоваться как Клиент. 2. Последовательно нажать на иконки вкладок внизу экрана: "Меню", "Рестораны", "Корзина", "Профиль". 3. Вернуться на вкладку "Меню". |
| Данные тестирования | Учетная запись клиента. |
| Ожидаемый результат | Переключение между вкладками происходит без видимых задержек. При нажатии на каждую иконку открывается соответствующий ей экран. Контент экрана отображается корректно. |
| Фактический результат | Соответствует |
| Предпосылки | Пользователь авторизован как Клиент. Приложение запущено на эмуляторе или устройстве. |
| Постусловия | Активна вкладка "Меню". |
| Статус (Pass/Fail) | Pass |
| Комментарии | Базовая проверка основной навигации приложения. |

**Тест-кейс И2**

|  |  |
| --- | --- |
| TestCase # | И2 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Mobile: Реакция кнопок +/- в корзине |
| Резюме испытания | Проверка визуальной реакции и обновления данных (количество, итоговая сумма позиции, общая сумма корзины) при нажатии кнопок увеличения и уменьшения количества товара в корзине. |
| Шаги тестирования | 1. Авторизоваться как Клиент. 2. Добавить товар в корзину. 3. Перейти в корзину. 4. Несколько раз нажать кнопку "+" для этого товара. 5. Наблюдать за изменением числа количества, суммы позиции и общей суммы корзины. 6. Несколько раз нажать кнопку "-". 7. Наблюдать за изменениями. |
| Данные тестирования | Любой товар в корзине. |
| Ожидаемый результат | 1. При каждом нажатии "+" количество товара увеличивается на 1. 2. При каждом нажатии "-" количество уменьшается на 1 (до минимума 1, при следующем нажатии товар удаляется -). 3. Сумма позиции и общая итоговая сумма корзины обновляются немедленно и корректно после каждого нажатия. 4. Кнопки визуально реагируют на нажатие. |
| Фактический результат | Соответствует |
| Предпосылки | Пользователь авторизован. В корзине есть хотя бы один товар. Логика изменения количества и пересчета суммы реализована и привязана к кнопкам. |
| Постусловия | Количество товара и итоговая сумма в корзине изменены. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка интерактивности и корректности обновления связанных данных в UI корзины. |

**Тест-кейс И3**

|  |  |
| --- | --- |
| TestCase # | И3 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Mobile: Читаемость описания блюда в карточке меню |
| Резюме испытания | Проверка читаемости текста описания блюда в карточке на главном экране меню, особенно для блюд с длинным описанием. Ожидается, что текст будет обрезан корректно, но тест проверяет возможную проблему с отображением. |
| Шаги тестирования | 1. Авторизоваться как Клиент. 2. Перейти в раздел "Меню". 3. Найти блюдо с относительно длинным описанием (более 2-3 строк при стандартном размере шрифта). 4. Оценить, как отображается текст описания в карточке. |
| Данные тестирования | Блюдо с длинным описанием (более 100 символов). |
| Ожидаемый результат | Текст описания должен быть читаемым. Если описание длинное, оно должно корректно обрезаться многоточием (или отображаться в заданное количество строк, например, 2 строки) без наложения на другие элементы или выхода за пределы карточки. |
| Фактический результат | Текст описания для длинных названий не обрезается корректно, а переносится на следующие строки, "раздвигая" карточку по высоте неравномерно по сравнению с другими карточками. |
| Предпосылки | Пользователь авторизован. В меню есть блюда с длинными описаниями. Используется стандартный ListView |
| Постусловия | Некорректное отображение элемента списка. |
| Статус (Pass/Fail) | Fail |
| Комментарии | Ошибка формирования карточки блюда (DataTemplate). Необходимо настроить свойство MaxLines="2" для Label с описанием. |

**Тест-кейс И4**

|  |  |
| --- | --- |
| TestCase # | И4 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Mobile: Отображение истории заказов |
| Резюме испытания | Проверка корректности отображения списка заказов в истории клиента: читаемость данных, отображение статуса. |
| Шаги тестирования | 1. Авторизоваться как Клиент, у которого есть несколько заказов в истории с разными статусами. 2. Перейти на вкладку "Профиль". 3. Просмотреть секцию "История заказов". |
| Данные тестирования | Учетная запись клиента с историей заказов (минимум 3-5 заказов с разными статусами). |
| Ожидаемый результат | Список заказов отображается. Для каждого заказа четко видны номер, дата, сумма, статус. Статусы, фон разного цвета в зависимости от статуса. Прокрутка списка плавная. |
| Фактический результат | Соответствует |
| Предпосылки | Пользователь авторизован. Есть история заказов. Реализована логика отображения статусов. |
| Постусловия | История заказов отображена корректно. |
| Статус (Pass/Fail) | Pass |
| Комментарии | Проверка визуального представления списка исторических данных. |

В ходе тестирования были выявлены и задокументированы дефекты различной степени критичности (включая ошибки в логике, UI и обработке исключений), большинство из которых были успешно устранены на этапе отладки и рефакторинга. Результаты тестирования подтверждают, что разработанные модули системы «Курочка и точка» соответствуют ключевым требованиям технического задания и демонстрируют стабильную работу в основных сценариях использования.