**Тестирование приложения Cardly**

**Общее о тестировании приложения Cardly**

**Цель тестирования:** обеспечение высокого качества мобильного приложения Cardly.  
Cardly - это мобильное приложение для коллекционирования и обмена виртуальными карточками между пользователями. Клиентская часть реализуется на Flutter, серверная - на Kotlin. Приложение предназначено для устройств на Android (начиная с версии 8). Тестирование направлено на выявление и устранение ошибок, связанных с ключевыми функциями, логикой пользовательского взаимодействия, отображением данных и стабильностью интерфейса.

**Основные модули приложения, подлежащие тестированию**

* Авторизация, регистрация и сброс пароля;
* Гостевой режим (ограниченный доступ без регистрации);
* Профиль пользователя (аватар, никнейм, статистика, настройки, избранные карты и достижения);
* Поиск пользователей;
* Коллекции и карточки (группировка, фильтрация, предпросмотр);
* Разбор карточек (обмен на валюту);
* Генерация уникальных карточек с помощью ИИ;
* Обмен карточками (быстрый и обычный, проверка активных обменов);
* Магазин (покупка валюты, наборов);
* Выполнение достижений и квестов (ежедневные/еженедельные);
* Получение уведомлений о событиях;
* Просмотр новостей;
* Жалобы на пользователей;
* Админ-панель.

**Слабые места тестирования**

* Ручная проверка обменов и генерации карточек;
* Отслеживание состояния карточек в инвентаре и истории обменов;
* Ограничение доступа для гостевого режима;
* Корректность уведомлений;
* Ограничений на повторное выполнение квестов.

**Техники тестирования**

* Ручное тестирование;
* Smoke-тестирование ключевых функций;
* Регрессионное тестирование;
* Проверка пользовательских сценариев.

**Инструмент ведения тестов**

* Google Таблицы

**План тестирования приложения Cardly**

**1. Процесс тестирования**

* Разработка тест-кейсов и чек-листов;
* Проведение ручного тестирования на Android-устройствах и эмуляторах разных версий Android (начиная с версии 8);
* Документация результатов, фиксация багов, обратная связь разработчикам.

**2. Проверка функциональности и работа с багами**

* Выполнение всех тест-кейсов;
* Сравнение результатов с ожидаемым поведением;
* Приём и верификация исправлений;
* Повторное тестирование затронутых сценариев.

**3. Актуализация тестовой документации**

* Обновление сценариев при изменении логики;
* Добавление новых кейсов по мере развития приложения.

**Процесс тестирования новых фич**

1. QA заранее пишет тест-кейсы по новой фиче на основе требований.
2. Разработчики предоставляют сборку с реализованной функцией.
3. QA запускает ручное тестирование на устройствах и проверяет работу компонентов.
4. Если найдены ошибки:
   * Пишется название;
   * Назначается приоритет;
   * Назначается серьёзность;
   * Пишется ожидаемый результат;
   * Пишется фактический результат;
   * Информация передаётся разработчикам.
5. После фиксов:
   * Выполняется повторное тестирование;
   * При необходимости - регрессия на затронутых модулях.
6. В завершении - smoke-тестирование ключевого функционала.
7. После успешной проверки QA подтверждает внедрение в основную ветку.

**Чек-лист написания проверок**

1. Предусловия;
2. Тестовые данные ;
3. Описание шагов;
4. Ожидаемый результат;
5. Ссылки на тз/дизайн, ответы от сервера по необходимости.