**Introduction**

Целью составления данного Test Plan является описание процесса тестирования приложения LazyCook. Документ позволяет получить представление о мероприятиях по тестированию проекта.

Целью тестирования приложения LazyCook является проверка его корректной работы, удобства приложения для пользователя, а также выявление пожеланий пользователя по расширению или исправлению существующего функционала.

**Test Items**

Объектом тестирования является приложение LazyCook, основным функционалом которого является хранение и поиск рецептов в персональной базе данных.

Данный проект должен обладать следующими атрибутами качества:

- Надежность (Reliability);

- Функциональные возможности (Functionality);

- Практичность (Usability).

**Risk Issues**

Пользователь приложение LazyCook может столкнуться со следующими рисками:

- Приложение может не запуститься под определенными операционными системами (Linux).

**Features to be Tested**

Тестированию в данном приложении подвергнутся:

- Функциональные требования приложения:

1) Возможность получения информации по любому запросу;

2) Возможность добавления нового рецепта.

3) Возможность удаления рецепта.

4) Возможность просмотра рецепта.

- Нефункциональные требования:

1) Надежность;

2) Удобный и простой интерфейс;

3) Практичность.

- Сценарии использования данного приложения

1) Добавление рецепта.

2) Удаление рецепта.

3) Просмотр рецепта.

4) Поиск рецепта по заданным условиям.

**Test Approach**

Тестирование будет производиться вручную, методом «неформального» тестирования c позиции конечного пользователя приложения.

**Pass / Fail Criteria**

Тестовые сценарии:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Purpose/Title | Scenario/Instructions | Expected Result |
| 1 | Проверка варианта использования «Добавление автора» основной поток событий. | 1) Запустить приложение;  2) Нажать кнопку «Add Author»;  3) В появившемся окне ввести имя «Anna»;  4) Нажать кнопку «Ok»  Приложение добавляет в базу данных имя нового автора. | В базу данных добавлено имя Anna. |
| 2 | Проверка варианта использования «Добавление автора» альтернативный поток событий. | 1) Запустить приложение;  2) Нажать кнопку «Add Author»;  3) В появившемся окне снова ввести имя «Anna»;  4) Нажать кнопку «Ok»  Приложение не добавляет в базу данных имя уже существующего автора. | Т.к. в базе данных уже есть пользователь с таким именем, оно не будет продублировано. |
| 3 | Проверка варианта использования «Добавление автора» поток ошибок. | 1) Удалить базу данных.  2) Запустить приложение;  3) Нажать кнопку «Add Author»;  4) В появившемся окне ввести имя «Anna»;  5) Нажать кнопку «Ok»  Из-за отсутствия базы данных приложение не сможет добавить новые данные. | Ввиду отсутствия базы данных, информация не будет добавлена. |
| 4 | Проверка варианта использования «Добавление типа блюда» основной поток событий. | 1) Запустить приложение;  2) Нажать кнопку «Add Type»;  3) В появившемся окне в первой строке ввести тип «Dessert»;  4) Во второй строке ввести вид подачи блюда «Serve hot»;  4) Нажать кнопку «Ok»  Приложение добавляет в базу данных имя нового типа блюда. | В базу данных добавлен тип Dessert с подачей Serve hot. |
| 5 | Проверка варианта использования «Добавление типа блюда» альтернативный поток событий. | 1) Запустить приложение;  2) Нажать кнопку «Add Type»;  3) В появившемся окне в первой строке снова ввести тип «Dessert»;  4) Во второй строке снова ввести вид подачи блюда «Serve hot»;  4) Нажать кнопку «Ok»  Приложение не добавляет в базу данных имя уже существующего типа блюда. | Т.к. в базе данных уже есть тип с таким именем, оно не будет продублировано. |
| 6 | Проверка варианта использования «Добавление типа блюда» поток ошибок. | 1) Удалить базу данных  2) Запустить приложение;  3) Нажать кнопку «Add Type»;  4) В появившемся окне в первой строке снова ввести тип «Dessert»;  5) Во второй строке снова ввести вид подачи блюда «Serve hot»;  6) Нажать кнопку «Ok»  Из-за отсутствия базы данных приложение не сможет добавить новые данные. | Ввиду отсутствия базы данных, информация не будет добавлена. |
| 7 | Проверка варианта использования «Добавление ингредиента» основной поток событий. | 1) Запустить приложение;  2) Нажать кнопку «Add Ingredient»;  3) В появившемся окне ввести имя «Milk»;  4) Нажать кнопку «Ok»  Приложение добавляет в базу данных имя нового ингредиента. | В базу данных добавлен ингредиент Milk. |
| 8 | Проверка варианта использования «Добавление ингредиента» альтернативный поток событий. | 1) Запустить приложение;  2) Нажать кнопку «Add Ingredient»;  3) В появившемся окне снова ввести имя «Milk»;  4) Нажать кнопку «Ok»  Приложение не добавляет в базу данных имя уже существующего ингредиента. | Т.к. в базе данных уже есть ингредиент с таким именем, оно не будет продублировано. |
| 9 | Проверка варианта использования «Добавление ингредиента» поток ошибок. | 1) Удалить базу данных.  2) Запустить приложение;  3) Нажать кнопку «Add Ingredient»;  4) В появившемся окне ввести ингредиент «Milk»;  5) Нажать кнопку «Ok»  Из-за отсутствия базы данных приложение не сможет добавить новые данные. | Ввиду отсутствия базы данных, информация не будет добавлена. |
| 10 | Проверка функционального требования «Возможность добавлять рецепт» | 1) Запустить приложение;  2) Нажать кнопку «Add Recipe»;  3) В появившемся окне ввести\выбрать необходимые поля;  4) Нажать кнопку «Ok» | Открывается основное окно, в котором в списке рецептов находится и новый рецепт. |
| 11 | Проверка функционального требования «Возможность просмотра рецепта» | 1. Запустить приложение; 2. Из списка рецептов выбрать любой и дважды нажать на него; | В окне появляется информация о данном рецепте. |
| 12 | Проверка функционального требования «Возможность удаления рецепта» | 1) Запустить приложение;  2) Из списка рецептов выбрать любой и дважды нажать на него;  3) Нажать на появившуюся кнопку «Delete Recipe»;  4) Подтвердить удаление; | Рецепт удален из базы данных и не отображается в главном окне. |
| 13 | Проверка функционального требования «Возможность поиска рецепта» | 1) Запустить приложение;  2) В верхнем левом углу выбрать критерий, по которому производится поиск;  3) В строке поиска ввести слово или его часть, по которому будет произведен поиск;  4) Нажать кнопку «Find»;  5) Если результаты не удовлетворяют запрос, нажать кнопку «Reset» и ввести другое слово\комбинацию или изменить критерий поиска; | В главном окне отображаются все рецепты, соответствующие данному запросу. |
| 14 | Проверка нефункционального требования «Удобный и простой интерфейс» | 1) Запустить приложение;  2) Осуществить нажатия на различные части приложения, тем самым проверить удобство работы с приложением. | Приложение удобно и просто в управлении. |
| 15 | Проверка нефункционального требования «Безопасность» | 1. Запустить приложение; 2. Проверить антивирусом на наличие вредоносных программ; | Приложение не содержит вирусов. |
| 16 | Проверка нефункционального требования «Надежность» | 1) Запустить приложение;  2) Выполнить любое из вышеперечисленных действий и получить гарантированный результат; | Приложение работает надежно. |

**Conclusion**

Результатом тестирования должно являться подтверждение выполнения приложением заявленных атрибутов качества. Представленные тесты покрывают функциональные и нефункциональные требования, а также заявленные варианты использования данного приложения.