**Введение**

Этот документ представляет собой заключительный этап разработки пользовательского приложения Resource Manager - тестирование.

Главной задачей данного этапа является проверка разработанного приложения на соответствие заявленному функционалу и атрибутам качества, а также поиск возможных скрытых неисправностей или недоработок, упущенных на стадии разработки. По результатам проведённого тестирования будет сделан вывод о готовности приложения для использования.

**Объекты тестирования**

Объектом тестирования является приложение Resource Manager. Основной задачей данного приложения является распределение заданного пользователем ресурса. Также в приложении имеется необходимый функционал поддержки, выполняющий второстепенные задачи.

К приложению заявлены требования по следующим атрибутам качества:

* Функциональность;
* Надёжность;
* Практичность.

**Риски**

Основным риском является отказ работоспособности приложения при следующих условиях:

* Использование приложения на операционных системах, не поддерживающих библиотеки кроссплатформенного языка Java.
* Использование приложения на чрезмерно устаревших машинах с точки зрения производительности.
* Отсутствие требуемых механизмов Java, являющихся основой для запуска и работы приложения.

**Аспекты тестирования**

Тестирование будет осуществляться над следующим списком возможностей и атрибутов:

* Сформулированные функциональные требования приложения:

1. Возможность создания нового расчёта
2. Возможность сохранить расчёт.
3. Возможность загрузить созданный ранее расчёт
4. Возможность добавить/удалить элемент списка расчёта
5. Возможность завершения работы приложения
6. Возможность пользовательского редактирования таких полей как: название ресурса, общее количество ресурсов, требуемое количество ресурсов и описание требуемого элемента списка расчёта

* Сформулированные нефункциональные требования приложения:

1. Точность: в приложении недопустимы никакие ошибки или погрешности в результатах: оставшееся количество ресурсов должно быть равно разности общего количества ресурсов и задействованных.
2. Стабильность: Приложение должно обеспечивать полный рабочий цикл, начиная с получения задачи и ввода данных, вплоть до получения результата и завершения работы приложения: пользователь запускает программу, создаёт задачу, решает её с помощью приложения и завершает работу.
3. Работоспособность под системами Windows, Linux: приложение запускает и обеспечивает выполнение функционала перечисленного в пункте «Функциональные требования».

**Подходы к тестированию**

Тестирование осуществляется практическим испытанием приложения, путём имитации стандартных рабочих условий, в которых в роли рядового пользователя выступает тестировщик.

**Тестовые сценарии**

Для тестирования определены следующие сценарии:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Purpose/Title | Scenario/Instructions | Expected Result |
| 1 | Тестирование варианта использования «Создание нового расчёта» | 1) Запустить приложение;  2) Нажать кнопку «Add»  3) Нажать кнопку «New»;  Приложение после выполнения шага 2 добавит один элемент в список, а после выполнения шага 3 обнулит его, что будет соответствовать новому расчёту | После нажатия кнопки «New» список элементов обнулится |
| 2 | Тестирование варианта использования «Сохранение расчёта»  Основной поток событий | 1) Запустить приложение;  2) Нажать кнопку «Add»;  3) Нажать кнопку «Save».  4) Ввести “TEST” в поле названия файла и подтвердить сохранение.  5) Снова нажать кнопку «Save».  В полученном окне добавился элемент с именем “TEST”, это значит, что файл был успешно сохранён | Файл с именем TEST появится в списке сохранённых файлов |
| 3 | Проверка варианта использования «Сохранение расчёта» альтернативный поток событий | 1) Запустить приложение;  2) Нажать кнопку «Add»;  3) Нажать кнопку «Save».  4) Ввести “TEST” в поле названия файла и подтвердить сохранение.  5) Снова нажать кнопку «Save».  6) Ввести “TEST” в поле названия файла и подтвердить сохранение.  Файл с таким именем уже имеется в списке, поэтому выведется диалоговое окно с запросом на перезапись | Появление диалогового окна с запросов на перезапись одноимённого файла из списка |
| 4 | Тестирование варианта использования «Загрузка расчёта» | 1) Запустить приложение;  2) 5 раз нажать кнопку «Add»;  3) Нажать кнопку «Save».  4) Ввести “TEST” в поле названия файла и подтвердить сохранение.  5) Нажать кнопку «New».  6) Нажать кнопку «Load»;  7) Выбрать “TEST” из списка и подтвердить загрузку.  После выполнения шага 5, элементы добавленные в список на шаге 2 будут удалены, а после выполнения шага 7 они снова появятся в списке. | Удалённые элементы восстановятся в таком количестве в каком были добавлены |
| 5 | Тестирование варианта использования «Добавление элемента в список» | 1) Запустить приложение;  2) Нажать кнопку «Add»;  По выполнению шага 2 в списке появится один новый элемент. | Новый элемент в списке расчёта |
| 6 | Тестирование варианта использования «Удаление элемента из списка» | 1) Запустить приложение;  2) Нажать кнопку «Add»;  3) Нажать на элементе кнопку в виде креста;  По выполнению шага 2 в списке появится один новый элемент, по выполнению шага три этот элемент удалится. | Элемент удалится из списка |
| 7 | Проверка варианта использования «Завершение работы приложения» | 1) Запустить приложение;  2) Нажать кнопку «Exit»  Приложение закроется | Окно приложения закроется |
| 8 | Проверка варианта использования «Редактирование полей панели состояния» | 1) Запустить приложение;  2) Ввести на панели состояния любую строку в качестве названия ресурса;  3) Перевести курсор и ввести число.  В результате на панели состояния поле названия ресурсов будет хранить введённую информацию, а поле баланса выведет введённое число | Сохранение введённой информации в полях панели состояния |
| 9 | Проверка варианта использования «Редактирование полей элемента списка» | 1) Запустить приложение;  2) Нажать кнопку «Add»;  3) Ввести на панели элемента любую строку в качестве описания варианта использования ресурса;  4) Перевести курсор и ввести число.  В результате поле описания элемента будет хранить введённую информацию, а поле баланса выведет введённое число со знаком минус. | Сохранение введённой информации в полях элемента |
| 10 | Проверка нефункционального требования «Точность» | 1) Запустить приложение;  2) Нажать кнопку «Add» 5 раз;  3) В числовое поле каждого из появившихся элементов ввести цифру 1  В результате поле баланса должно вывести сумму чисел соответствующих полей всех элементов списка со знаком минус, то есть значение -5 | Поле баланса на панели состояния будет содержать значение -5 |
| 11 | Проверка нефункционального требования «Стабильность» | 1)Запустить приложение  2)Осуществить сценарии 1-6,8,9? Исключая шаг 1 | Приложение работает нормально. Получены результаты поиска. |
| 12 | Проверка функционального требования «Работоспособность под системами Windows, Linux» | 1)Осуществить сценарии тестирования 1-11 под системами Windows и Linux | Приложение выполняет запросы, результаты сценариев тестирования идентичны. |

**Заключение**

Выполнение тестирования должно подтвердить работоспособность представленного приложения, а также подтвердить его соответствие заявленным требованиям и атрибутам качества, описанным на этапе проектирования приложения. Гарантия этого соответствия обеспечивается сформулированными сценариями тестирования.