**Тест-план**

**ПО “ListBoxer”**

# Введение

## Цели

Заказчику нужна программа для создания алфавитно-цифровых универсальных списков, прошедшая полный цикл ручного тестирования. Учитывая спецификацию ПО очень важно иметь такое же качество и дизайн самого ПО.

План тестирования был создан для облегчения общения внутри членов команды. В настоящем документе описываются подходы и методологии, которые будут применяться к подразделению, интеграции и системному тестированию. Она включает в себя цели, обязанности по тестированию, критерии входа и выхода, сферу охвата, график основных этапов, критерии входа и выхода и подход. Этот документ четко определил, какими будут результаты испытаний, а также то, что считается входящим и выходящим из сферы применения.

## Члены команды

|  |  |
| --- | --- |
| **Имя** | **Роль** |
| ... | Разработчик |
| ... | Менеджер проекта |
| Рафкат Галеев | Тестировщик |

# Структура

Начальная фаза включает в себя все требования, предъявленные в спецификации. Эти и любые другие требования, которые будут включены, будут проверены. В конце фазы 1 будут получены следующие результаты:

1. Будет разработан план ручного тестирования
2. Будет сохранен и извлечен тест-план
3. Использование тест-плана во время ручного тестирования
4. Результаты теста
5. Просмотр и анализ результатов тестирования

Как только будут получены результаты 1 фазы тестирования, команда определит следующие цели для 2 фазы.

# Риски

Были выявлены следующие риски и определены соответствующие меры по смягчению их воздействия на проект. Влияние (или серьезность) риска зависит от того, как проект будет затронут, если риск будет инициирован. Триггер - это то, какая веха или событие может привести к тому, что риск превратится в проблему, требующую решения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Риск | Влияние | Триггер | План Смягчения Последствий |
| 1 | Размывание поставленных дат проекта | Высокое | Задержки с датой внедрения | Каждая итерация, функциональность будет тщательно контролироваться. Приоритеты будут установлены и обсуждены заинтересованными сторонами. В случае непредвиденных обстоятельств, возможно, придется отодвинуть дату. |
| 2 | Изменения в функциональности могут свести на нет уже написанные тесты и мы можем потерять уже написанные тестовые кейсы | Высокое | Потеря всех тестовых кейсов | Экспорт данных до любого обновления по мере необходимости и повторный импорт после обновления. |
| 3 | Еженедельная доставка невозможна, так как разработчик работает вне сайта | Средняя | Продукт не был доставлен вовремя |  |
| 4 |  |  |  |  |

# Тестовый Подход

Проект использует гибкий подход, с еженедельными итерациями. В конце каждой недели требования, определенные для этой итерации, будут доставлены команде и будут проверены.

## Автоматизация тестирования

Автоматизированные модульные тесты являются частью процесса разработки, но в настоящее время автоматизированные функциональные тесты не планируются.

# Среда тестирования

Для тестирования ПО необходима операционная система Linux или Windows.

# Основные Этапы / Конечные Результаты

## Расписание тестов

Первоначальный график испытаний:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задача** | **Начало** | **Окончание** | **Попытка** | **Комментарии** |
| Планирование теста | 27.04.2020 | 27.04.2020 | 1 |  |
| Проверка необходимых документов | 27.04.2020 | 27.04.2020 | 4 |  |
| Создание начальных тестовых оценок | 27.04.2020 | 27.04.2020 | 1 |  |
| Подготовка новых тестовых ресурсов | 27.04.2020 | 27.04.2020 | 2 |  |
| Первое развертывание в тестовой среде | 27.04.2020 | 27.04.2020 | 3 |  |
| Функциональное тестирование-1-ая итерация | 27.04.2020 | 27.04.2020 | 1 | ПО не удалось установить |
| Функциональное тестирование-2-ая итерация | 27.04.2020 | 27.04.2020 | 1 | Ошибка в исполняемом файле |
| Разрешение окончательных дефектов и окончательное тестирование сборки | 27.04.2020 | 27.04.2020 | 1 |  |

## Результаты

|  |  |  |
| --- | --- | --- |
| **Результаты** | **Для** | **Дата** |
| Тест-план | Менеджер проекта; Группа тестирования |  |
| Результаты теста | Менеджер проекта |  |
| Отчет о состоянии испытаний | Менеджер, Руководитель |  |

**История версий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Версия # | Дата | Автор | Итоговые изменения | Обзор комментариев |
| 1.98 | 27.04.2020 | Рафкат Галеев | Первый тест-кейс |  |
|  |  |  |  |  |

**Цель:** Скачать программу, запустить исполняемый файл, проверить функциональность ПО

**Тип тестирования:** Функциональное тестирование

**Предварительные требования:** ОС Linux/Windows

**Предварительные условия:** Компьютер, Ноутбук, Тестировщик

**Тест-кейсы**

|  |  |  |  |
| --- | --- | --- | --- |
| Число | Действие | Ожидаемый результат | Результат тестирования |
| **1** | **Основное** | | |
| **1.1** | **Запустить программу** | | |
|  | Подготовка:   1. Скачать программу |  |  |
| 1.1.1 | Шаги:   1. Запустить установщик программы ListBoxer | Запуск установщика программы ListBoxer | Установщик запускается |
| 1.1.2 | Шаги:   1. Установить программу ListBoxer | Установка программы ListBoxer | Файл для установки SETUP.LST не найден |
| 1.1.3 | Шаги:   1. Запустить программу | Запуск программы | - |
| **1.2** | **Сворачивание-разворачивание программы** | | |
| 1.2.1 | Шаги:   1. Свернуть программу 2. Удостовериться, что программа свернута | Программа свернута | - |
| 1.2.2 | Шаги:   1. Развернуть программу 2. Удостовериться, что программа открылась | Программа развернута | - |
| **2** | **Ввод данных, режимы работы** | | |
| **2.1** | **Режимы работы** | | |
| 2.1.1 | Шаги:   1. Нажать на checkbox для смены режима ввода | Работа checkbox’а. Смена режима ввода | - |
| **2.2** | **Ввод данных** | | |
|  | Шаги:   1. Ввести данные 2. Нажать на клавишу “Enter” или на кнопку “Add to list” | Данные добавлены в список | - |
| 2.2.1 | Шаги:   1. Выбрать алфавитный режим 2. Ввести текстовые данные | Отображение текстовых данных в программе | - |
| 2.2.2 | Шаги:   1. Выбрать цифровой режим 2. Ввести численные данные | Отображение численных данных в программе | - |
| 2.2.2 | Шаги:   1. Выбрать смешанный режим 2. Ввести данные смешанного типа | Отображение данных смешанного типа в программе | - |
| **2.3** | **Исправность ввода в разных режимах** | | |
| 2.3.1 | Шаги:   1. Выбрать режим алфавитный 2. Проверка разрешенного диапазона ввода от “а” до “z”, от “A” до “Z” 3. Проверка количества символов в строке. Допустимо 8 | Все разрешенные символы отображены в программе. В строке 8 символов. | - |
| 2.3.2 | Шаги:   1. Выбрать цифровой режим 2. Проверка разрешенного диапазона ввода чисел от 0 до 9999 3. Проверка количества символов в строке. Допустимо 8 | Все разрешенные символы отображены в программе. В строке 8 символов. | - |
| 2.3.3 | Шаги:   1. Выбрать смешанный режим 2. Проверка разрешенного диапазона ввода от “а” до “z”, от “A” до “Z” 3. Проверка разрешенного диапазона ввода чисел от 0 до 9999 4. Проверка количества символов в строке. Допустимо 8 | Все разрешенные символы отображены в программе. В строке 8 символов. | - |
| **3** | **Фильтры, сортировка** | | |
| **3.1** | **Фильтры** | | |
| 3.1.1 | Шаги:   1. Выбрать алфавитный режим 2. Нажать на выпадающий список фильтров 3. Удотовериться в наличии вариантов фильтров: “<none>”,”All”,”a-m”,”n-z” 4. Ввести текстовые данные различных видов 5. Отфильтровать текстовые данные | Наличие всех перечисленных вариантов фильтров, текст отфилтрован исправно | - |
| 3.1.2 | Шаги:   1. Выбрать цифровой режим 2. Нажать на выпадающий список фильтров 3. Проверить наличии враинатов фильтров:”<none>”,”All”,”0-100”,”101-200”,”201-300” и “301-9999” 4. Ввести численные данные 5. Отфильтровать числовые данные | Наличие всех перечисленных варинатов филтрации, текст отфильтрован исправно | - |
| 3.1.3 | Шаги:   1. Выбрать смешанный режим 2. Нажать на выпадающий список фильтров 3. Удостовериться в наличии всех вариантов филтрации обоих режимов 4. Ввести данные 5. Отфильтровать данные | Наличие всех вариантов фильтрации, данные отфильтрованы исправны | - |
| **3.2** | **Сортировка** |  | |
| 3.2.1 | Шаги:   1. Ввести данные 2. Выбрать сортировку по возрастанию | Данные отсортированы по возрастанию | - |
| 3.2.2 | Шаги:   1. Ввести данные 2. Выбрать сортировку по убыванию | Данные отсортированы по убыванию | - |
| **4** | **Проверка кнопок программы** | | |
| **4.1** | **Удаление введенной информации** | | |
| 4.1.1 | Шаги:   1. Ввести данные 2. Нажать на кнопку “Clear list” | Введенная информация удалена | - |
| **4.2** | **Отмена последнего действия** |  | |
| 4.2.1 | Шаги:   1. Ввести данные 2. Нажать на кнопку “Undo” | Отмена введенной информации | - |