|  |  |
| --- | --- |
|  | **Тестовый документ** |

## **Аннотация теста**

|  |  |
| --- | --- |
| **Название проекта** | SellPaperDemoExam |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Гребенюк Андрей |
| **Дата(ы) теста** | **09.03.24** |

## **Расшифровка тестовых информационных полей:**

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Название тестируемого проекта |
| **Рабочая версия** | Версия проекта/программного обеспечения (первый тест считается 1.0). |
| **Имя тестирующего** | Имя того, кто проводил тесты |
| **Дата(ы) теста** | Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста. |
| **Тестовый пример #** | Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например,‘TC\_UI\_1′ означает‘user interface test case #1′ ( ТС\_ПИ\_1: тестовый случай пользовательского интерфейса#1) |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет. |
| **Заголовок/название теста** | Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем. |
| **Краткое изложение теста** | Описание того, что должен достичь тест. |
| **Этапы теста** | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея. |
| **Тестовые данные** | Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа. |
| **Ожидаемый результат** | Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране. |
| **Фактический результат** | Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста. |
| **Предварительное условие** | Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая. |
| **Постусловие** | Каким должно быть состояние системы после выполнения теста? |
| **Статус**  *(Зачет/Незачет)* | Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено. |
| **Примечания/комментарии** | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами). |

## Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** | Тест1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | button2\_Click\_PageChange |
| **Краткое изложение теста** | Проверка функционала кнопки перехода на предыдущую страницу, с случае если предыдущей страницы нет. |
| **Этапы теста** | form.button4\_Click(null, null); Нажатие кнопки перехода  Assert.AreEqual("1", form.currentPage.ToString()); Проверка соответсвия индекса страницы с ожидаемым результатом |
| **Тестовые данные** | Идекс страницы формы |
| **Ожидаемый результат** | Индекс страницы должен быть равен 1. |
| **Фактический результат** | Кнопка перехода работает корректно, не совершая переход |
| **Статус** | Зачет |
| **Предварительное условие** | Существование формы AgentList  Наличие доступа к данным из БД  Наличие в БД данных об агенте и его продажах |
| **Постусловие** | Форма AgentList осталась на той же странице |

## Тестовый пример #2:

|  |  |
| --- | --- |
| **Тестовый пример #** | Тест2 |
| **Приоритет тестирования** | Низкий |
| **Заголовок/название теста** | Exit\_Click\_ClosesForm |
| **Краткое изложение теста** | Проверка работоспособности кнопки отмены создания агента |
| **Этапы теста** | addForm.button2\_Click(null, null); Нажатие кнопки отмены  Assert.IsFalse(addForm.Visible); Проверка формы AddAgent на видимость |
| **Тестовые данные** | Форма AddAgent |
| **Ожидаемый результат** | Форма AddAgent закрылась |
| **Фактический результат** | Форма AddAgent закрылась |
| **Статус** | Зачет |
| **Предварительное условие** | Существование формы AddAgent |
| **Постусловие** | Форма AddAgent закрыта |

## Тестовый пример #3:

|  |  |
| --- | --- |
| Тестовый пример **#** | Тест3 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | TextBox1\_TextChanged\_Test |
| **Краткое изложение теста** | Проверить работоспособность поля ввода |
| **Этапы теста** | form.textBox1.Text = "100"; Ввод текста в поле ввода  form.textBox1\_TextChanged(null, null); Запуск метода для фильтрации по тексту  Assert.AreEqual("100", form.textBox1.Text); Проверка текста в поле на соответсвие изначальным данным |
| **Тестовые данные** | Вводные данные для текстового поля |
| **Ожидаемый результат** | Форма выводит агентов по фильтру текста 100 и в поле остается это значение |
| **Фактический результат** | Форма выводит агентов по фильтру текста 100 и в поле остается это значение |
| **Статус** | Зачет |
| **Предварительное условие** | Существование формы AgentList  Наличие доступа к данным из БД  В БД есть данные об агентах и их продажах |
| **Постусловие** | Форма вывела агентов по фильтру и в поле ввода введено 100. |

## Тестовый пример #4:

|  |  |
| --- | --- |
| Тестовый пример **#** | Тест4 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | button3\_Click\_PageChange |
| **Краткое изложение теста** | Проверить работоспособность кнопки перехода на следующую страницу формы |
| **Этапы теста** | form.button3\_Click(null, null); Нажатие кнопки перехода  Assert.AreEqual("2", form.currentPage.ToString()); Проверка соответсвия индекса страницы с ожидаемым результатом |
| **Тестовые данные** | Идекс страницы формы |
| **Ожидаемый результат** | Индекс страницы должен быть равен 2. |
| **Фактический результат** | Кнопка перехода работает, но не пропускает на следующую страницу из-за остутствия достаточного количества агентов в БД |
| **Статус** | Незачет |
| **Предварительное условие** | Существование формы AgentList  Наличие доступа к данным из БД  В БД есть данные об агентах и их продажах в количестве больше 3. |
| **Постусловие** | Форма AgentList перешла на следующую страницу |
| **Примечания/комментарии** |  |

## Тестовый пример #5:

|  |  |
| --- | --- |
| **Test Case #** | Тест5 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | TestAgentRetrieval |
| **Краткое изложение теста** | Проверить наличия агентов в БД |
| **Этапы теста** | using (SellPaper\_Test3Entities db = new SellPaper\_Test3Entities()) { } подключение к базе данных  var agents = db.Agent.ToList(); конвертация списка агентов в List  Assert.IsNotNull(agents);  Assert.IsTrue(agents.Count > 0); Проверка на наличие агентов |
| **Тестовые данные** | БД SellPaper\_Test |
| **Ожидаемый результат** | В БД есть данные о агентах и их кол-во не равно или меньше нуля |
| **Фактический результат** | В БД есть данные о агентах и их кол-во не равно или меньше нуля |
| **Статус** | Зачет |
| **Предварительное условие** | Есть доступ к БД  В БД есть данные об агентах |
| **Постусловие** | Получено подтверждение о наличии агентов |
| **Примечания/комментарии** |  |