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

## **Аннотация теста**

|  |  |
| --- | --- |
| **Название проекта** | MarksGenerator |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Мустафа Алибеков |
| **Дата(ы) теста** | 02.07.22 |

## **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 #** | TestMethodSymbols |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка ведомости Посещения |
| **Краткое изложение теста** | Функция должна возвращать случайное количество пропусков, больше -1  и меньше 11. |
| **Этапы теста** | 1. Передать функции передать функции ФИО, дату рождения и группу студента 2. Вызвать функцию 3. Проверить возвращенное значение на соответствие ожидаемому результату |
| **Тестовые данные** | Mustafa Alibekov Magomedeminovich  195  2000 |
| **Ожидаемый результат** | Вызов функции присылает корректное количество пропусков |
| **Фактический результат** | Вызов функции прислал корректное количество пропусков |
| **Статус** | Зачет |
| **Предварительное условие** | Должны быть переданы данные студента |
| **Постусловие** | Функция возращает количество пропусков и заканчивает работу |
| **Примечания/комментарии** |  |

## **Тестовый пример #2:**

|  |  |
| --- | --- |
| **Тестовый пример #** | TestMethodGetStudNumberReturn(True)(False) |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка Правильности Создания Студенческого билета |
| **Краткое изложение теста** | Метод получает данные студента и возвращает студенческий билет |
| **Этапы теста** | 1. Метод Получает данные студента 2. Метод возвращает студенческий билет 3. Тест проверяет на соответствие полученные данные из метода |
| **Тестовые данные** | Mustafa Alibekov Magomedeminovich  195  2000 |
| **Ожидаемый результат** | Метод правильно сформирует студенческий билет, Первый тест покажет true, второй false так как ожидаемые данные будут изменены |
| **Фактический результат** | Метод правильно сформулировал студенческий билет. Первый тест показал true, второй false так как ожидаемые данные были изменены |
| **Статус** | Зачет |
| **Предварительное условие** |  |
| **Постусловие** |  |
| **Примечания/комментарии** |  |