**Валидация, Верификация и Тестови план**

**Проект ПТС с номера 3 и 2**

**Ревизионен лист**

|  |  |  |
| --- | --- | --- |
| **Релийз No.** | **Дата** | **Описание** |
| Rev. 0 | 5/30/00 |  |
| Rev. 1 | 4/12/02 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Валидация, Верификация и Тестови план**

**СЪДЪРЖАНИЕ**

Page #

1.0 Обща информация 1-1

1.1 Цел 1-1

1.2 Обхват 1-1

1.3 Общ преглед на системата 1-1

1.5 Съкращения 1-1

2.0 Оценка на тестването 2-1

2.1 Матрица за проследяване на изискванията (Requirements Traceability Matrix) 2-1

2.2 Критерии за оценка на тест 2-1

3.0 Описание на тестовите случаи 3-1

3.1 Функционални тестове 3-1

3.1.х [Test Identifier] 3-1

 *Тестови / функционални връзки* 3-1

 *Средства за контрол* 3-1

 *Входни данни* 3-1

 *Изходни данни* 3-1

 *Процедура за тестване* 3-1

3.2 Модулни (Unit) тестове 3-1

3.2.х [Test Identifier] 3-1

 *Средства за контрол* 3-1

 *Входни данни* 3-2

 *Изходни данни* 3-2

 *Покритие на теста* 3-2

# Обща информация

## 1.1 Цел

Целта на плана за валидация е да се разгледат и потвърдят главните случаи свързани с потребителските истории.

## 1.2 Обхват

Да се достигне пълно покритие на логиката за потребителските истории

## 1.3 Общ преглед на системата

Целта на системата е да изчисли мерки на разсейване. Данните, върху които се прави анализа се взимат от предоставените ни файлове - Course A\_StudentsResults\_Year 1.xlsx, Course A\_StudentsResults\_Year 2.xlsx и Logs\_Course A\_StudentsActivities.xlsx.

## 1.5 Съкращения - Няма

# ОЦенка на тестването

## 2.1 Матрица за проследяване на изискванията (Requirements Traceability Matrix)

|  |  |  |  |
| --- | --- | --- | --- |
| Изисвакване № | Описание на изискването | Идентификатор на теста | Статус |
| 1 | Създай временен текстов файл, в който да се запишат данните от log файла | Модулен тест №1 | Pass |
| 2 | Изчисли стандартно отклонение | Модулен тест №2 | Pass |
| 3 | Извлечи данни от текстовия файл | Модулен тест №3 | Pass |

## 2.2 Критерии за оценка на тест

|  |  |
| --- | --- |
| 6 | Приложението да изчисли и визуализира размаха |
| 7 | Приложението да изчисли и визуализира стандартното отклонение |
| 8 | Приложението да изчисли и визуализира дисперсията |

# Описание на тестовите случаи

## 3.1 Функционални тестове

*Описват се всички функционални тестове, които ще се проведат. Всеки тест трябва да бъде в отделен раздел, 3.1.1 – 3.1.x.*

### 3.1.1 Неопределен брой user-и преглеждат дадени лекции, от които изчисляваме мерки на разсейване, но има грешно форматиране в log файла.

### *Тестови / функционални връзки*

Тест за потребителски истории 6,7,8

### *Средства за контрол*

Тестът бива извършен след като се избере конкретната опция от програмата.

### *Входни данни – log файл*

### *Изходни данни – При грешно форматиране се извежда съобщение на потребителя*

### *Процедура за тестване*

1. Натискане на бутона „Мерки на разсейване за прегледани лекции”
2. Отваряне, извличане на данните от log файла и записването им в текстов файл
3. Проверка ред по ред на текстовия файл и откриване на грешно форматиране

### 3.1.2 Неопределен брой user-и преглеждат дадени лекции, от които изчисляваме мерки на разсейване, но log файла е празен.

### *Тестови / функционални връзки*

Тест за потребителски истории 6,7,8

### *Средства за контрол*

Тестът бива извършен след като се избере конкретната опция от програмата.

### *Входни данни – log файл*

### *Изходни данни – При празен log файл се извежда съобщение на потребителя*

### *Процедура за тестване*

1. Натискане на бутона „Мерки на разсейване за прегледани лекции”
2. Отваряне, извличане на данните от log файла и записването им в текстов файл
3. Проверка за това дали новосъздадения текстов файл е празен

### 3.1.3 Неопределен брой user-и преглеждат дадени лекции, от които изчисляваме мерки на разсейване, но има прекалено малко записи в log файла.

### *Тестови / функционални връзки*

Тест за потребителски истории 6,7,8

### *Средства за контрол*

Тестът бива извършен след като се избере конкретната опция от програмата.

### *Входни данни – log файл*

### *Изходни данни – При прекалено малко записи се извежда съобщение на потребителя*

### *Процедура за тестване*

1. Натискане на бутона „Мерки на разсейване за прегледани лекции”
2. Отваряне, извличане на данните от log файла и записването им в текстов файл
3. Проверка на това дали има достатъчно данни в log файла, за извършването на съответните изчисления

## 

## 3.2 Модулни (Unit) тестове

*Описват се всички модулни тестове, които ще се проведат. Всеки тест трябва да бъде в отделен раздел, 3.2.1 – 3.2.x.*

### 3.2.1 Модулен тест №1

Име: CopyExcelTableToTempTextFile\_TempFilePath\_Test –

Целта на теста е да провери дали метода връща правилен Path към временния текстов файл

### *Средства за контрол*

В използваната среда трябва да изпълним тестове с цъкането на бутона “RunTests”

### *Входни данни – Пътят към временния текстов файл*

### *Изходни данни - Няма*

### *Покритие на теста – 100%*

### 3.2.2 Модулен тест №2

Име: FindStandartDeviation\_Test –

Целта на теста е да провери дали метода изчислява правилно стандартното отклонение

### *Средства за контрол*

В използваната среда трябва да изпълним тестове с цъкането на бутона “RunTests”

### *Входни данни – 1,2,3,4*

### *Изходни данни - Няма*

### *Покритие на теста – 100%*

### 3.2.3 Модулен тест №3

Име: ExtractDataFromTempTextFile\_Test –

Целта на теста е да провери дали метода правилно извлича данните от текстовия файл, който е създаден след прочитането на log файла

### *Средства за контрол*

В използваната среда трябва да изпълним тестове с цъкането на бутона “RunTests”

### *Входни данни – string:* "File: Лекция 8: Език за заявки SPARQL\nFile: Лекция 7: Език за заявки SPARQL"

### *Изходни данни - Няма*

### *Покритие на теста – 100%*