**Валидация, Верификация и Тестови план**

*Статистически анлиз*

**Ревизионен лист**

|  |  |  |
| --- | --- | --- |
| **Релийз No.** | **Дата** | **Описание** |
| Rev. 0 | 11/04/22 | Добавяне на функционални и модулни тестове |
| Rev. 1 | 17/04/22 | Добавяне на изчислено покритие на тестове |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**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

3.1.1 Валидни входни данни [TC01] 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

3.1.2 Невалиден път към файл [TC02] 3-2

 *Тестови / функционални връзки* 3-2

 *Средства за контрол* 3-2

 *Входни данни* 3-2

 *Изходни данни* 3-2

 *Процедура за тестване* 3-2

3.1.3 Невалидни входни данни[TC03] 3-3

 *Тестови / функционални връзки* 3-3

 *Средства за контрол* 3-3

 *Входни данни* 3-4

 *Изходни данни* 3-4

 *Процедура за тестване* 3-4

3.1.4 Валидни входни данни [TC04] 3-5

 *Тестови / функционални връзки* 3-5

 *Средства за контрол* 3-5

 *Входни данни* 3-5

 *Изходни данни* 3-5

 *Процедура за тестване* 3-5

3.1.5 Множество от един елемент[TC05] 3-6

 *Тестови / функционални връзки* 3-6

 *Средства за контрол* 3-6

 *Входни данни* 3-6

 *Изходни данни* 3-6

 *Процедура за тестване* 3-6

3.1.6 Изследване на празно множество [TC06] 3-7

 *Тестови / функционални връзки* 3-7

 *Средства за контрол* 3-7

 *Входни данни* 3-7

 *Изходни данни* 3-8

 *Процедура за тестване* 3-8

3.2 Модулни (Unit) тестове 3-9

3.2.1 GetDispersion\_ListOfData\_Returns315\_25 [4.1] 3-9

 *Средства за контрол* 3-9

 *Входни данни* 3-9

 *Изходни данни* 3-9

 *Покритие на теста* 3-9

3.2.2 GetDispersion\_OneElementList\_ThrowsArgumentExeption [4.2] 3-9

 *Средства за контрол* 3-9

 *Входни данни* 3-9

 *Изходни данни* 3-9

 *Покритие на теста* 3-10

3.2.3 GetDispersion\_EmptyList\_ThrowsArgumentExeption [4.3] 3-10

 *Средства за контрол* 3-10

 *Входни данни* 3-10

 *Изходни данни* 3-10

 *Покритие на теста* 3-10

3.2.4 GetDispersion\_NullReference\_ThrowsArgumentExeption [4.4] 3-10

 *Средства за контрол* 3-10

 *Входни данни* 3-10

 *Изходни данни* 3-11

 *Покритие на теста* 3-11

3.2.5 GetStandardDeviation\_ListofData\_Returns26\_25 [4.5] 3-11

 *Средства за контрол* 3-11

 *Входни данни* 3-11

 *Изходни данни* 3-11

 *Покритие на теста* 3-11

3.2.6 GetStandardDeviation\_OneElementList\_ThrowsArgumentExeption [4.6] 3-11

 *Средства за контрол* 3-11

 *Входни данни* 3-12

 *Изходни данни* 3-12

 *Покритие на теста* 3-12

3.2.7 GetStandardDeviation\_EmptyList\_ThrowsArgumentException [4.7] 3-12

 *Средства за контрол* 3-12

 *Входни данни* 3-12

 *Изходни данни* 3-12

 *Покритие на теста* 3-12

3.2.8 GetStandardDeviation\_NullReference\_ThrowsArgumentExeption [4.8] 3-12

 *Средства за контрол* 3-13

 *Входни данни* 3-13

 *Изходни данни* 3-13

 *Покритие на теста* 3-13

# Обща информация

## 1.1 Цел

Целта на тестването е да се валидират и верифицират елементи изчисляващи дисперсия и стандартно отклонение от бизнес изискване Изчисляване на мерки на разсейване за редактирано Wiki

## 1.2 Обхват

Обхвата на тестване включва функционални тестове по метода на бялата кутия и модулни тестове на функциите. Няма да се провежда автматизирано тестване на потребителския интерфейс, тъй като тестването му е обхванато от функционалните тестове

## 1.3 Общ преглед на системата

Системата извършва статистическа обработка на логове от система за онлайн обучение. Статистическите обработки включват: извършване на честотен анализ, изчисляване на мерки на централна тенденция, мерки на разсейване на брой редактирани Уики от потребител и корелацинен анализ между множествата на брой редактирани Уики от потребител и брой каачени файлове от потребител.

## 1.5 Съкращения

# ОЦенка на тестването

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

<TraceabilityMatrix.xlsx>

## 2.2 Критерии за оценка на тест

Приложението да изчисли и визуализира дисперсията.

Приложението да изчисли и визуализира стандартното отклонение.

# Описание на тестовите случаи

## 3.1 Функционални тестове

### 3.1.1 Валидни входни данни-дисперсия [TC01]

### *Тестови / функционални връзки*

Изчисляване на дисперсия-TC01

### *Средства за контрол*

Тестът се контролира с ръчни средства.

### *Входни данни*

test\_scores.xlsx

test\_logs.xlsx

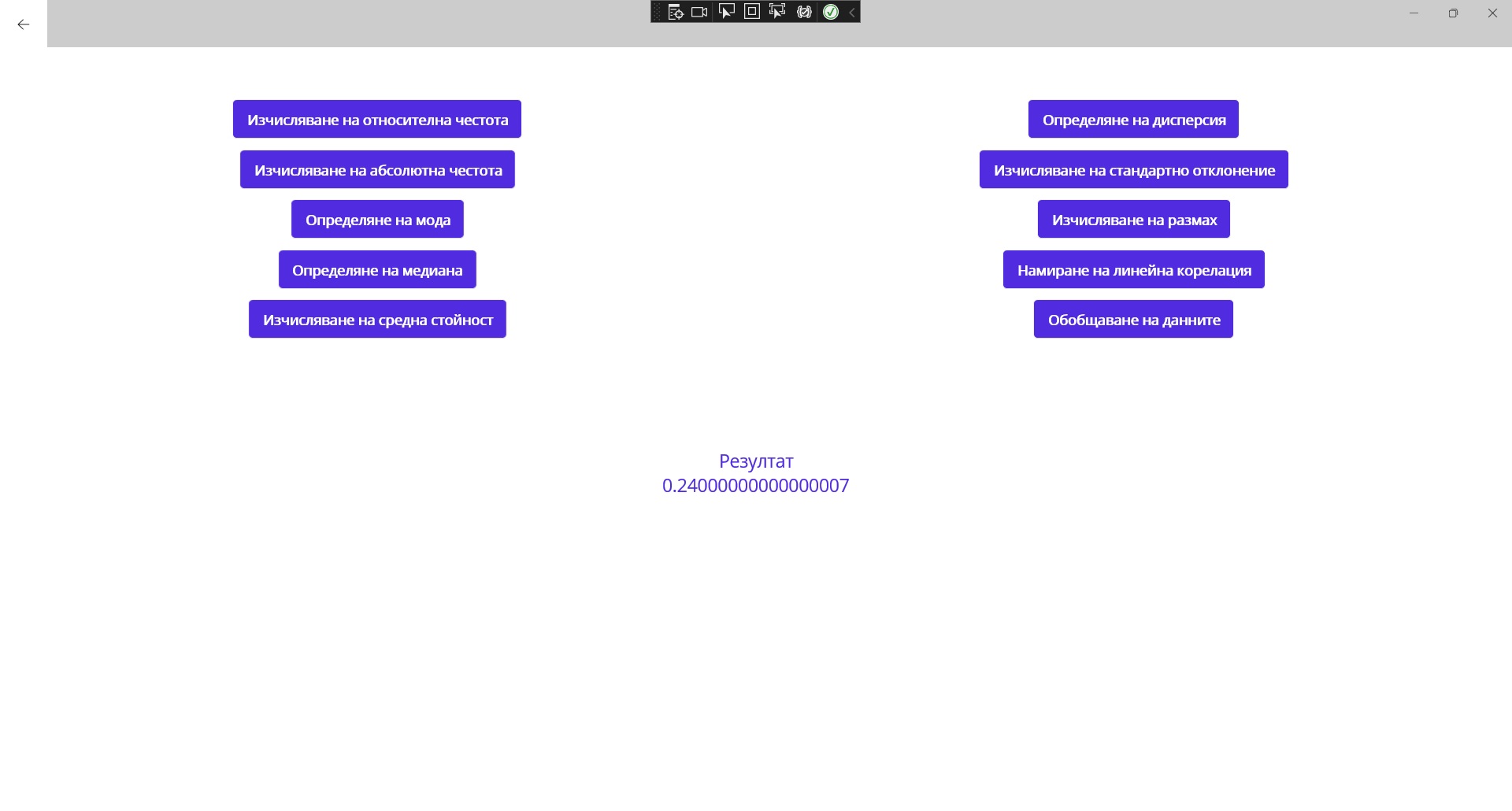
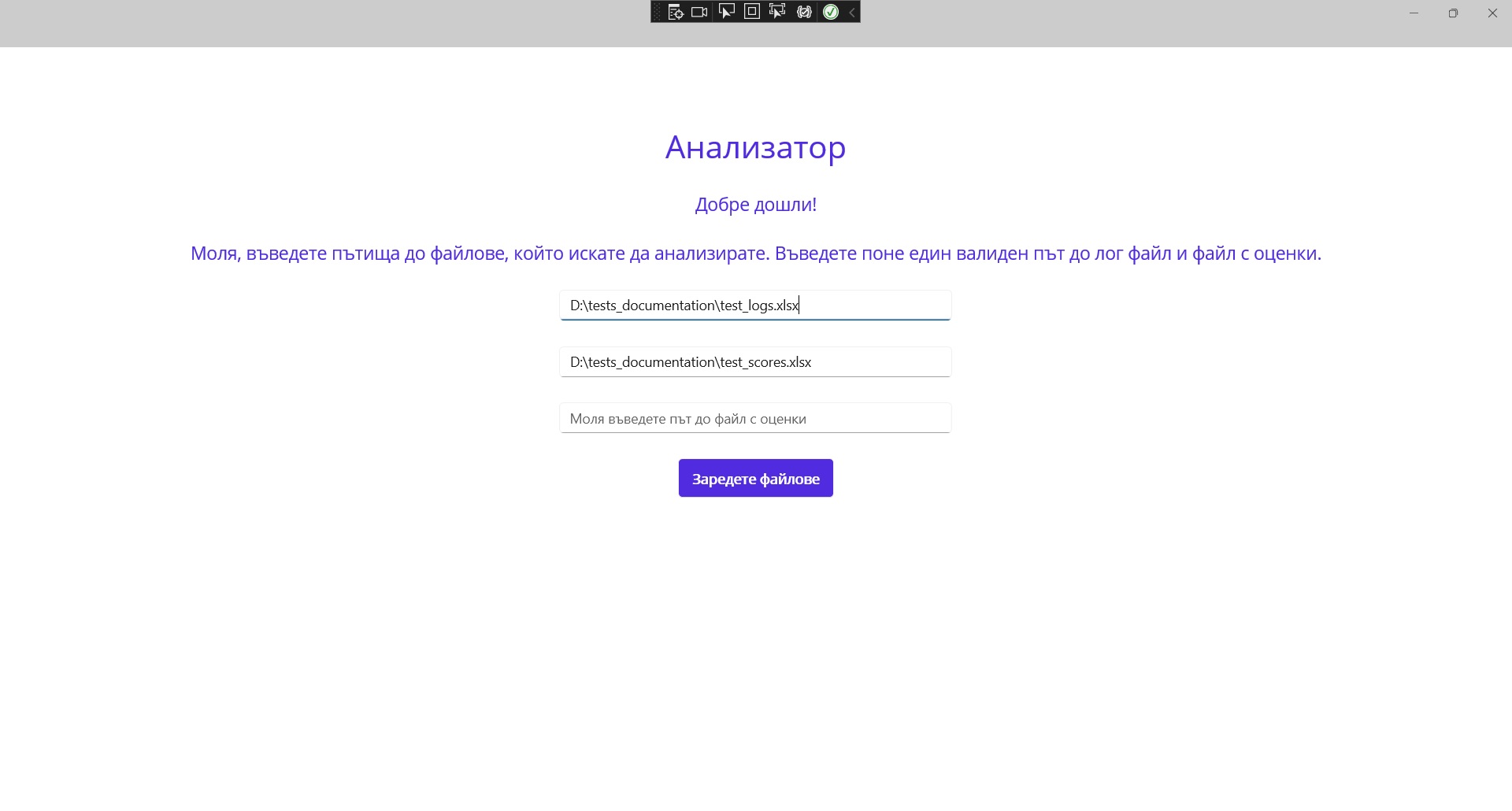
При тези входни файлове изследваното множество е 1,1,2,2.

### *Изходни данни*

0.24

### *Процедура за тестване*

1. Въвеждане на път към файлове
2. Зареждане на файлове
3. Натискане на бутон за изчисляване на дисперсия



### 3.1.2 Изследване на множество от един елемент-дисперсия [TC02]

### *Тестови / функционални връзки*

Изчисляване на стандартно отклонение-TC05

### *Средства за контрол*

Тестът се контролира с ръчни средства.

### *Входни данни*

test\_scores.xlsx

one\_log.xlsx

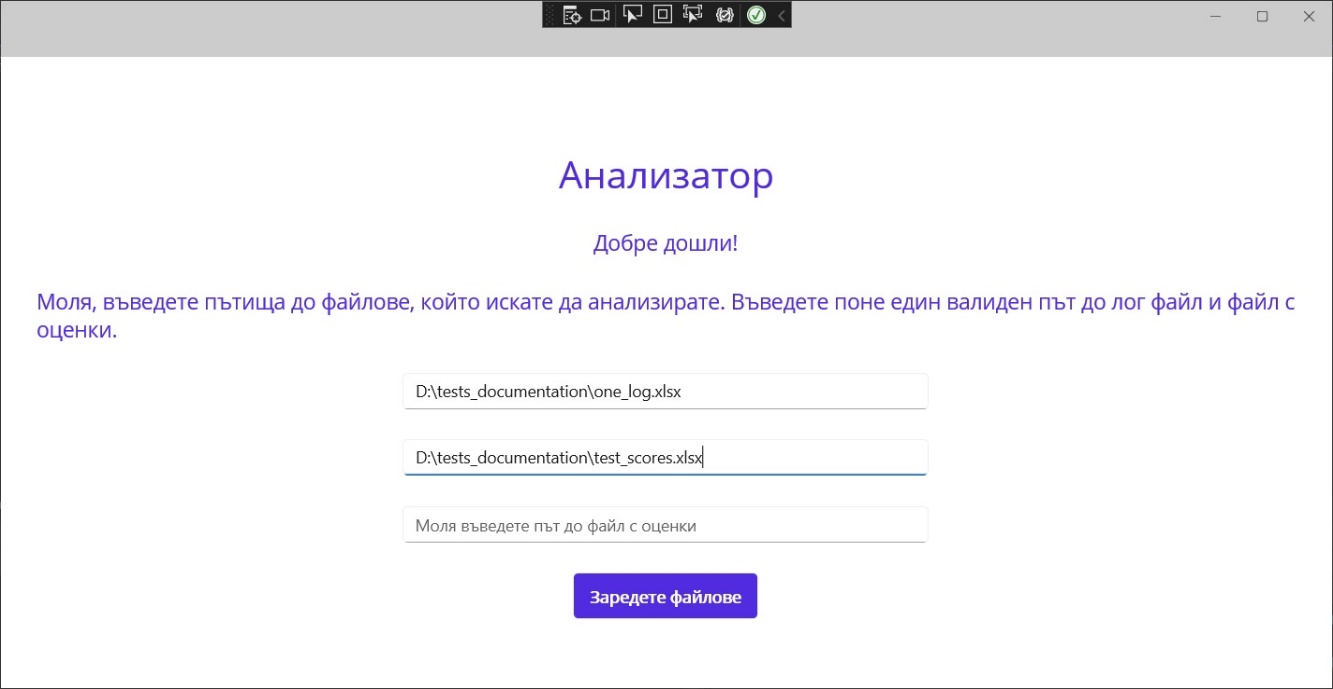
При тези входни файлове изследваното множество е 1.

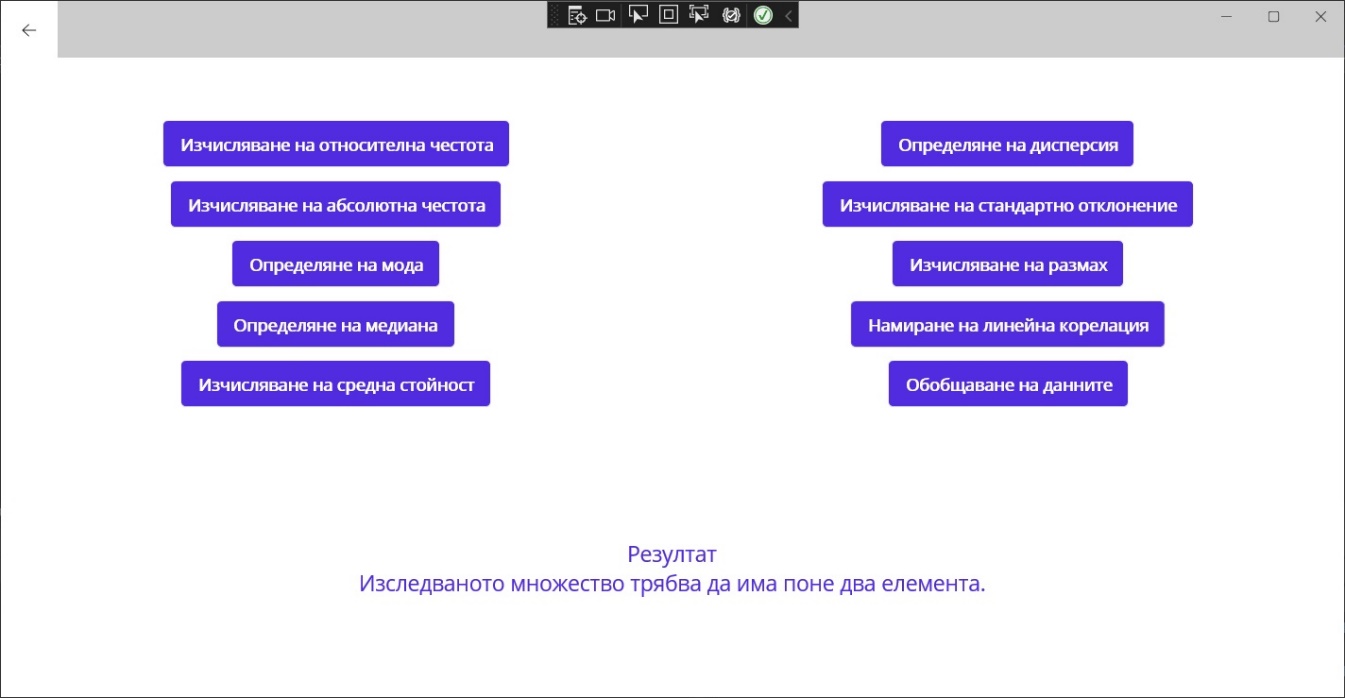
### *Изходни данни*

Изследваното множество трябва да има поне два елемента.

### *Процедура за тестване*

1. Въвеждане на път към файлове
2. Зареждане на файлове
3. Натискане на бутон за изчисляване на дисперсия





### 3.1.3 Невалидни входни данни-дисперсия[TC03]

### *Тестови / функционални връзки*

Изчисляване на дисперсия-TC03

### *Средства за контрол*

Тестът се контролира с ръчни средства.

### *Входни данни*

invalid\_scores.xlsx

invalid\_logs.xlsx

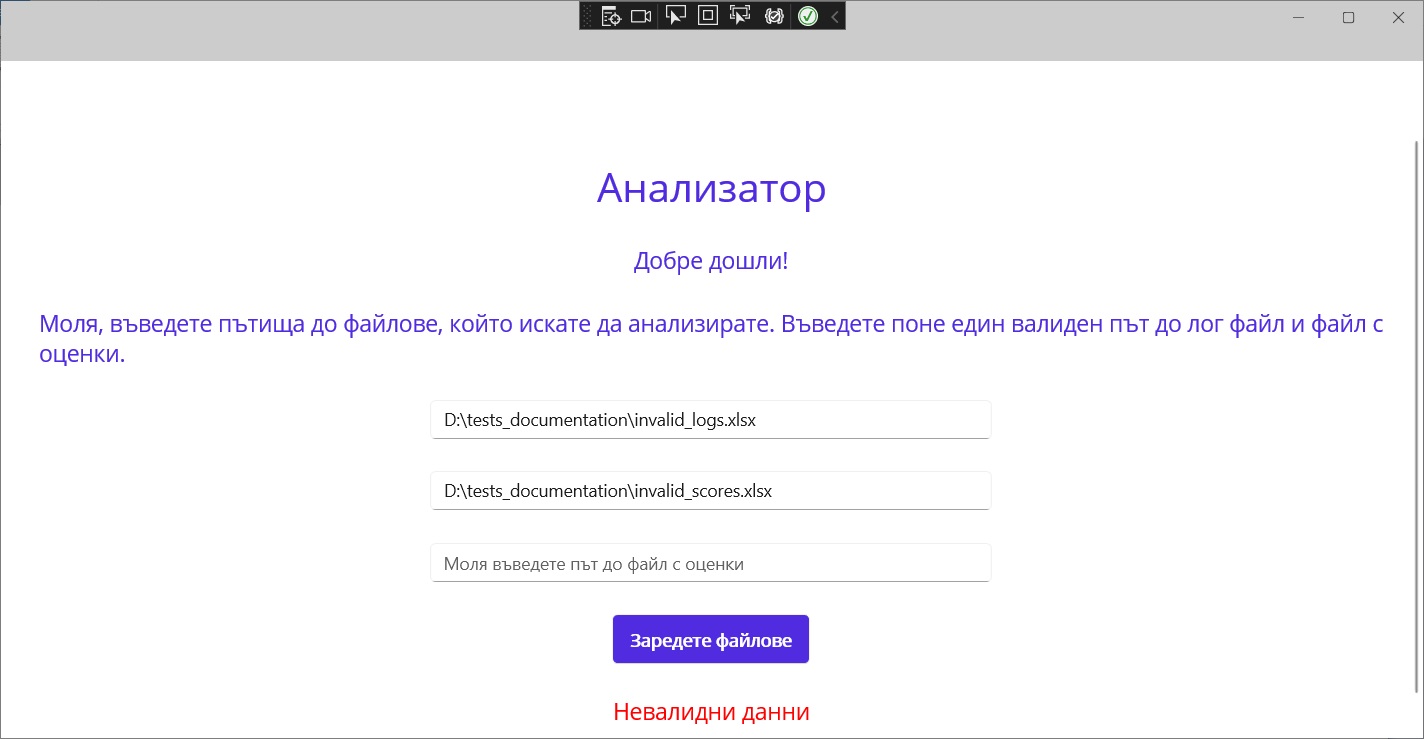
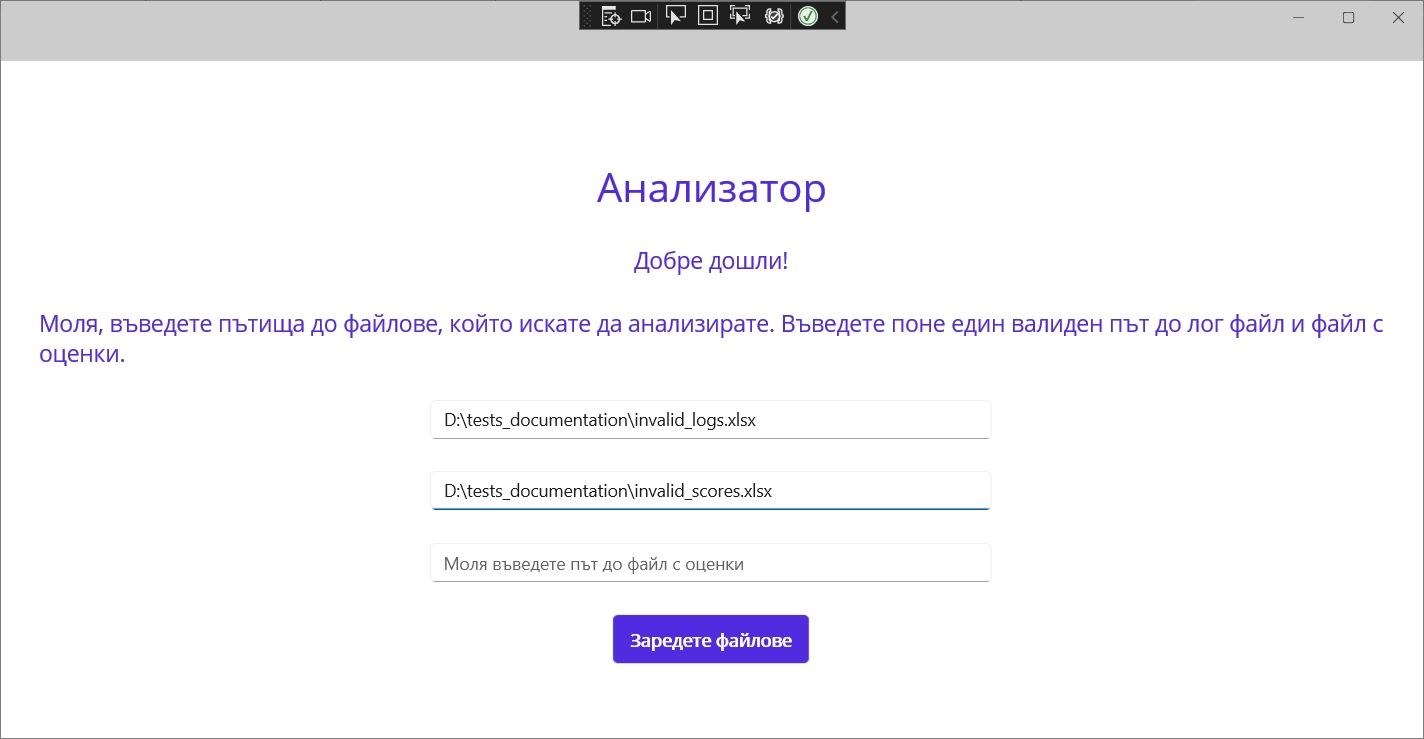
При тези входни файлове няма изследвано множество.

### *Изходни данни*

Невалидни данни

### *Процедура за тестване*

1. Въвеждане на път към файлове
2. Зареждане на файлове



### 3.1.4 Валидни входни данни-стандартно отклонение [TC04]

### *Тестови / функционални връзки*

Изчисляване на стандартно отклонение-TC04

### *Средства за контрол*

Тестът се контролира с ръчни средства.

### *Входни данни*

test\_scores.xlsx

test\_logs.xlsx

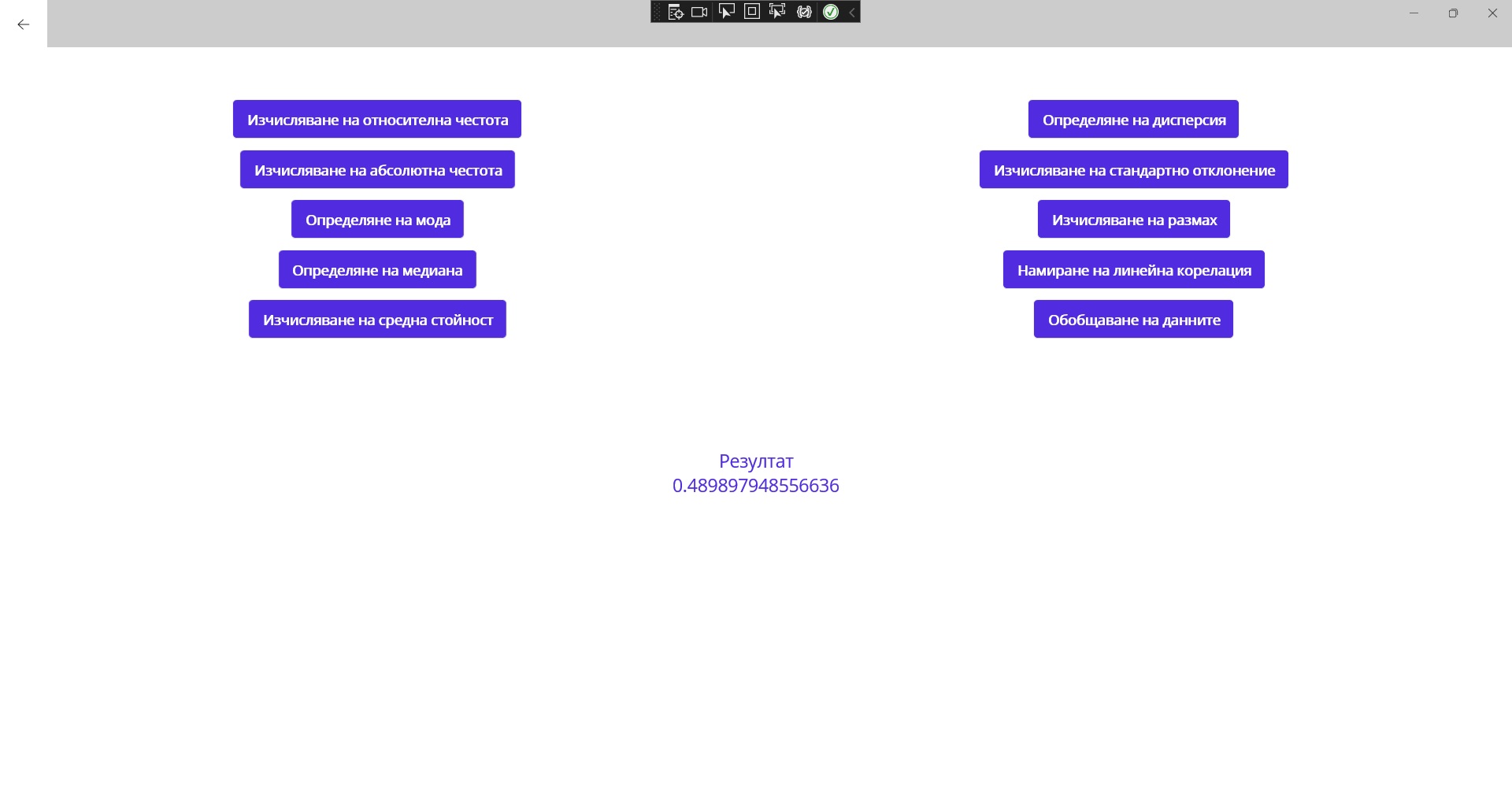
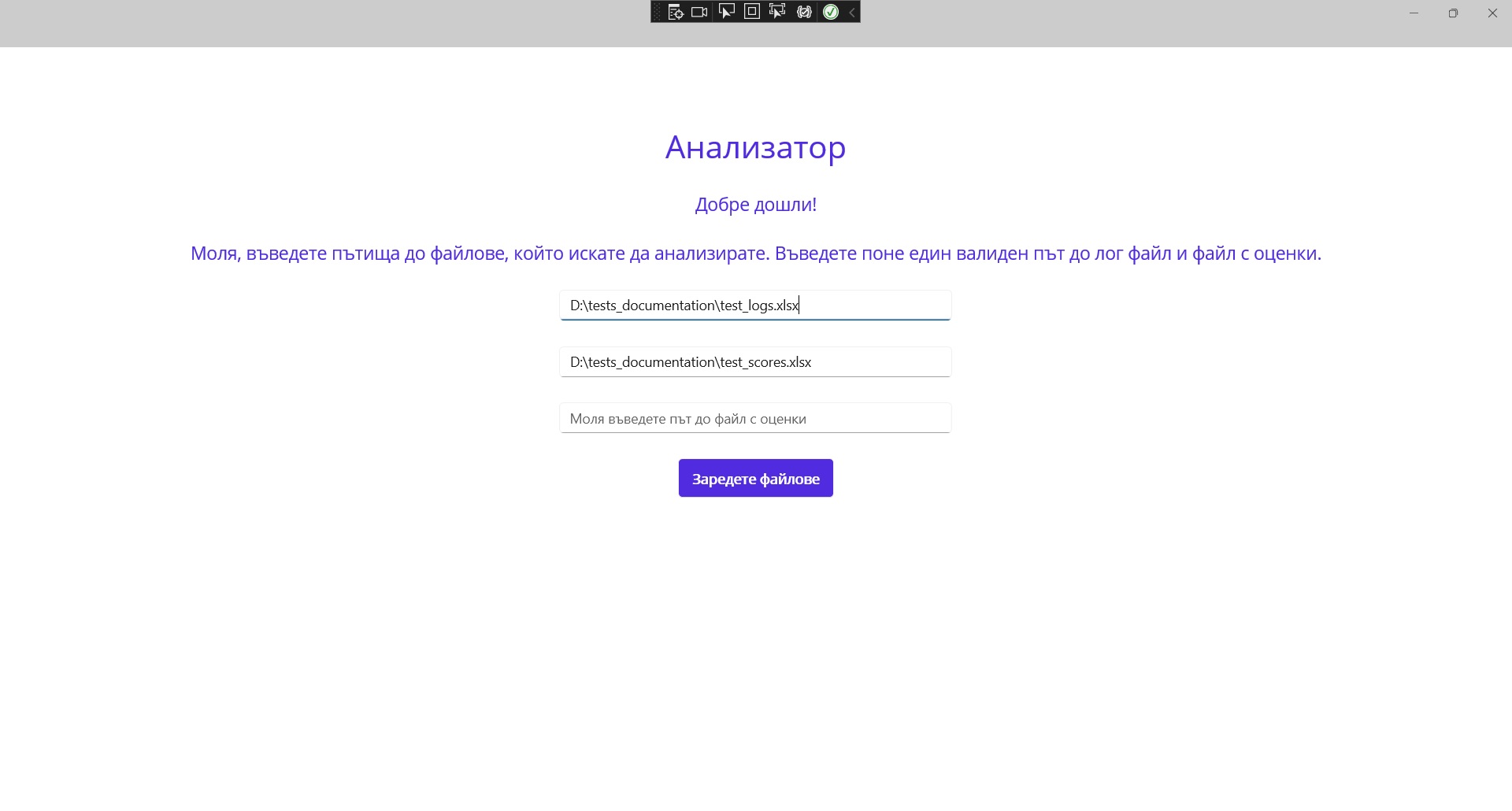
При тези входни файлове изследваното множество е 1,1,2,2.

### *Изходни данни*

0.489

### *Процедура за тестване*

1. Въвеждане на път към файлове
2. Зареждане на файлове
3. Натискане на бутон за изчисляване на стандартно отклонение



### 3.1.5 Множество от един елемент-стандартно отклонение [TC05]

### *Тестови / функционални връзки*

Изчисляване на стандартно отклонение-TC05

### *Средства за контрол*

Тестът се контролира с ръчни средства.

### *Входни данни*

test\_scores.xlsx

one\_log.xlsx

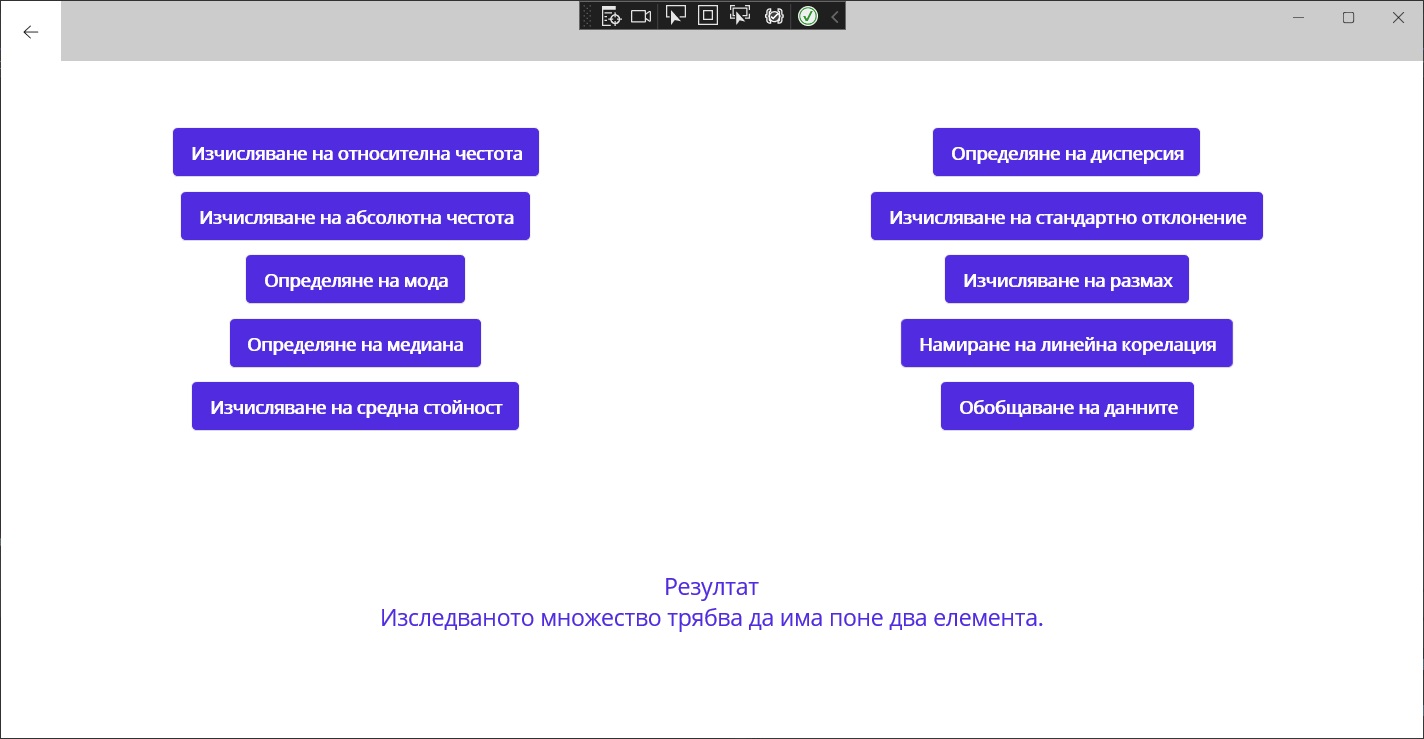
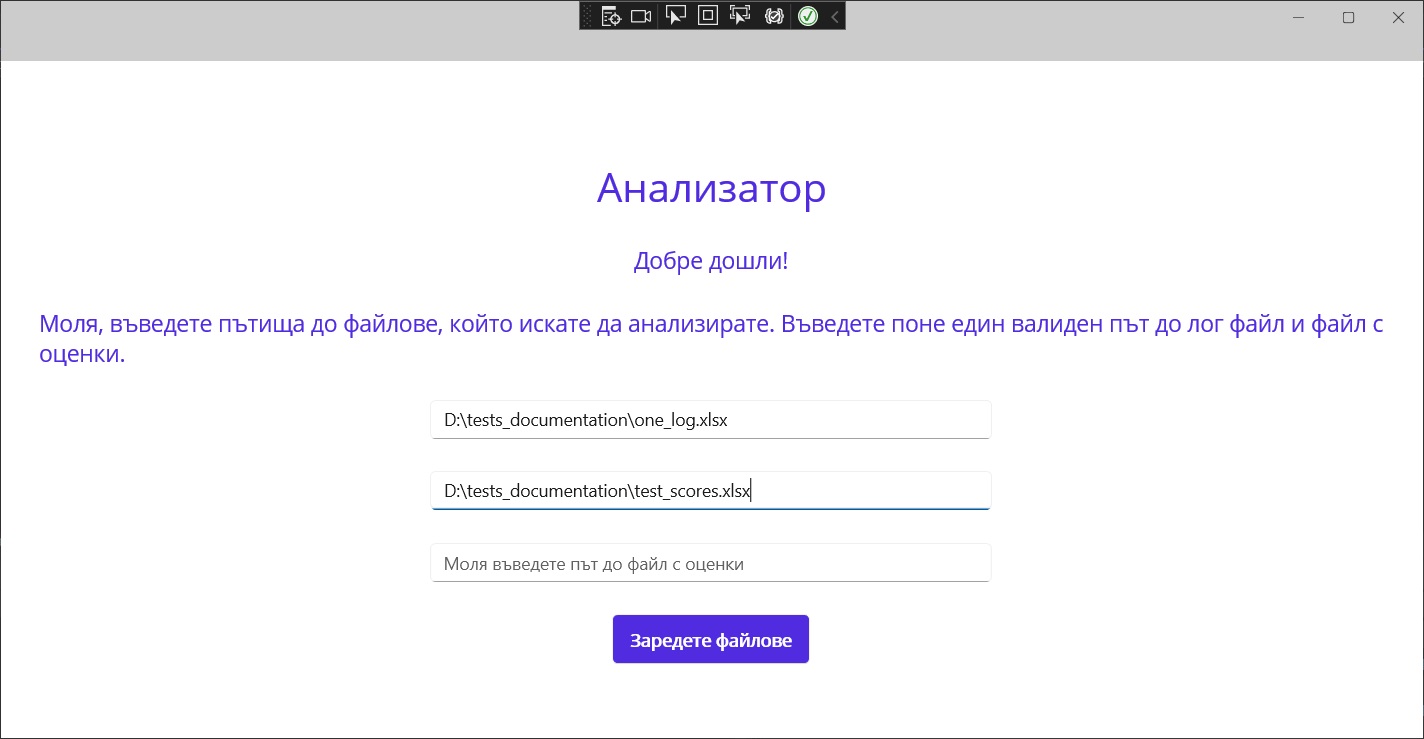
При тези входни файлове изследваното множество е 1.

### *Изходни данни*

Изследваното множество трябва да има поне два елемента.

### *Процедура за тестване*

1. Въвеждане на път към файлове
2. Зареждане на файлове
3. Натискане на бутон за изчисляване на стандартно отклонение



### 3.1.6 Изследване на празно множество-стандартно отклонение [TC06]

### *Тестови / функционални връзки*

Изчисляване на стандартно отклонение-TC04

### *Средства за контрол*

Тестът се контролира с ръчни средства.

### *Входни данни*

test\_scores.xlsx

empty\_log.xlsx

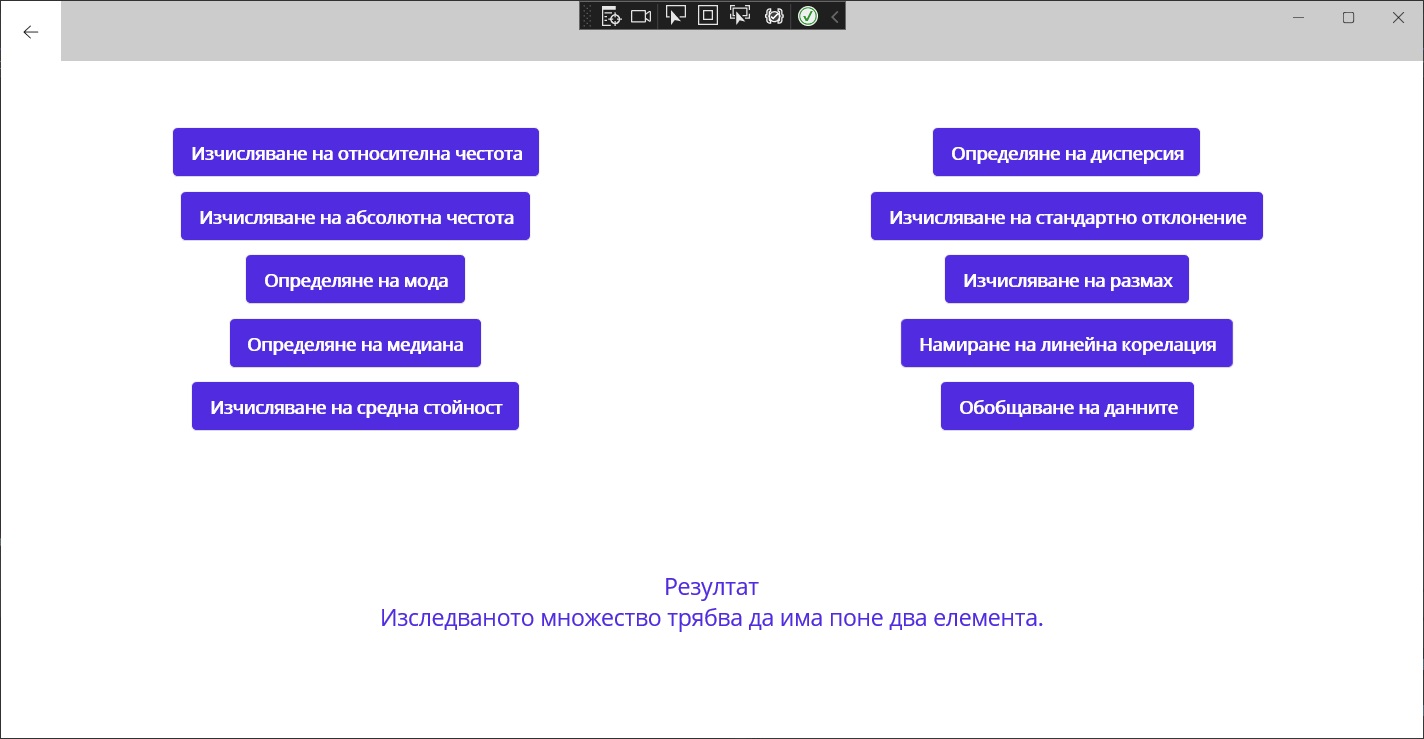
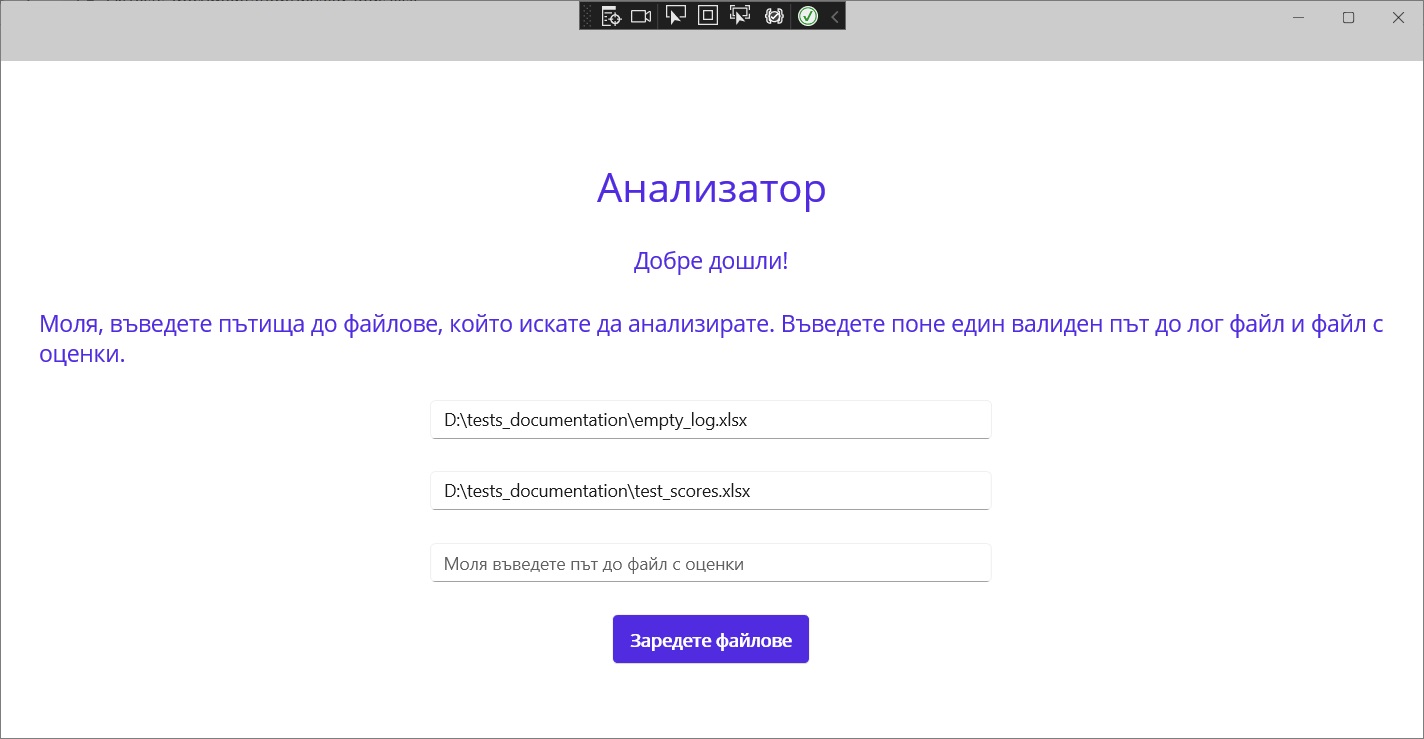
При тези входни файлове се изследва празно множество.

### *Изходни данни*

Изследваното множество трябва да има поне два елемента.

### *Процедура за тестване*

1. Въвеждане на път към файлове
2. Зареждане на файлове
3. Натискане на бутон за изчисляване на стандартно отклонение



## 3.2 Модулни (Unit) тестове

### 3.2.1 GetDispersion\_ListOfData\_Returns315\_25 [4.1]

Тестът проверява дали методът за изчисляване на дисперсия GetDispersion работи правилно при подаден списък с данни. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int> testData = new() { 6, 7, 12, 49 };

### *Изходни данни*

Очакван резултат-315.25

Полчучен резултат-315.25

### *Покритие на теста*

Statement coverage-85, 71%

Decision coverage-50%

Condition coverage-50%

### 3.2.2 GetDispersion\_OneElementList\_ThrowsArgumentExeption [4.2]

Тестът проверява дали методът за изчисляване на дисперсия GetDispersion предизвиква грешка при подаден списък с един елемент. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int> testData = new() { 6 };

### *Изходни данни*

Очакван резултат-ArgumentExeption

Полчучен резултат-ArgumentExeption

### *Покритие на теста*

Statement coverage-42, 86%

Decision coverage-50%

Condition coverage-50%

### 3.2.3 GetDispersion\_EmptyList\_ThrowsArgumentExeption [4.3]

Тестът проверява дали методът за изчисляване на дисперсия GetDispersion предизвиква грешка при подаден празен списък. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int> testData = new();

### *Изходни данни*

Очакван резултат-ArgumentExeption

Получен резултат-ArgumentExeption

### *Покритие на теста*

Statement coverage-42, 86%

Decision coverage-50%

Condition coverage-50%

### 3.2.4 GetDispersion\_NullReference\_ThrowsArgumentExeption [4.4]

Тестът проверява дали методът за изчисляване на дисперсия GetDispersion предизвиква грешка при референция към null обект. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int>? testData = null;

### *Изходни данни*

Очакван резултат-ArgumentExeption

Получен резултат-ArgumentExeption

### *Покритие на теста*

Statement coverage-42, 86%

Decision coverage-50%

Condition coverage-50%

### 3.2.5 GetStandardDeviation\_ListofData\_Returns26\_25 [4.5]

Тестът проверява дали методът за изчисляване на стандартно отклонение GetStandardDeviation работи правилно при подаден списък с данни.. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int> testData = new() { -7, -13, -28, -74 };

### *Изходни данни*

Очакван резултат- 26.253571

Почучен резултат- 26.253571

### *Покритие на теста*

Statement coverage-88, 88%

Decision coverage-50%

Condition coverage-50%

### 3.2.6 GetStandardDeviation\_OneElementList\_ThrowsArgumentExeption [4.6]

Тестът проверява дали методът за изчисляване на стандартно отклонение GetStandardDeviation предизвиква грешка при подаден списък с един елемент. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int> testData = new() { 3 };

### *Изходни данни*

Очакван резултат- ArgumentExeption

Почучен резултат- ArgumentExeption

### *Покритие на теста*

Statement coverage-88, 88%

Decision coverage-50%

Condition coverage-50%

### 3.2.7 GetStandardDeviation\_EmptyList\_ThrowsArgumentException [4.7]

Тестът проверява дали методът за изчисляване на стандартно отклонение GetStandardDeviation предизвиква грешка при подаден празен списък. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int> testData = new();

### *Изходни данни*

Очакван резултат- ArgumentExeption

Почучен резултат- ArgumentExeption

### *Покритие на теста*

Statement coverage-33, 33%

Decision coverage-50%

Condition coverage-50%

### 3.2.8 GetStandardDeviation\_NullReference\_ThrowsArgumentExeption [4.8]

Тестът проверява дали методът за изчисляване на стандартно отклонение GetStandardDeviation предизвиква грешка при референция към null обект. Метода е част от функционалност от бизнес изискване 4- Изчисляване на мерки на разсейване за редактирано Wiki.

### *Средства за контрол*

Тестът се контролира с рамката за автоматично тестване NUnit. Покритието на теста са изчислява ръчно.

### *Входни данни*

List<int>? testData = null;

### *Изходни данни*

Очакван резултат- ArgumentExeption

Почучен резултат- ArgumentExeption

### *Покритие на теста*

Statement coverage-33, 33%

Decision coverage-50%

Condition coverage-50%