# 

# 

# 

**Populat.io**

**TEST REPORT**

Version 1.1

29/03/2018

**VERSION HISTORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | Team | 15-03-2018 |  |  | Concept of Test plan |
| 1.1 | Team | 29-03-2018 |  |  | Test report after iteration one |
| 2.0 | Team | 23-05-2018 |  |  | Test report at the end of iteration 2 |
|  |  |  |  |  |  |

**TABLE OF CONTENTS**

[**FUNCTIONAL REQUIREMENTS**](#_w6rvx2oyhgxl) **4**

[**TEST CASE 1**](#_ihh2hqjrhrul) **4**

[**TEST CASE 2**](#_xvlobvb7pgyi) **4**

[**TEST CASE 3**](#_2ps4t2stalqi) **5**

[**TEST CASE 4**](#_h96nlm69od0y) **5**

[**TEST CASE 5**](#_q5dozgizaw72) **6**

[**TEST CASE 6**](#_1vuv2jnc7ho4) **7**

[**NON-FUNCTIONAL REQUIREMENTS**](#_7tfrubs0khti) **7**

[**TEST CASE 7**](#_r3ggmsui3ib1) **7**

[**TEST CASE 8**](#_nigiutoygaa2) **8**

# 

## FUNCTIONAL REQUIREMENTS

### TEST CASE 1

Test case description: Loading data from a file.

Preconditions: Program is running and files are provided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User clicks on the import a file button. |  | System shows a choose file dialog. | ✔  File dialog is shown. |
| 2. | User chooses a file to import. | Available city files (csv). | System loads data from the file to the screen. | ✔  File data is imported and charts are updated. |
| 3. | User compares shown data with data from chosen file. |  | Shown data matches data from chosen file. | ✔  Data in charts matches data from file. |

### TEST CASE 2

Test case description: Manually changing simulation parameters.

Preconditions: Program is running and a file has already been loaded.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User clicks on the change parameters button |  | System prompts a new window with different input fields for the different parameters such as population number, birth rate, death rate, immigration and emigration rate, average age of the population. | ✖  Not implemented as of iteration one |
| 2. | User types in the input fields and clicks on the Confirm/Ok/Apply button to apply the changes. | Input data | System applies the changes and closes the ‘change parameters’ window. | ✖ |

### TEST CASE 3

Test case description: Simulating the growth of a city’s population.

Preconditions: Program is running and a file has successfully been loaded.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User should change the year from 2017 to the year that he/she requires to be simulated to and user changes the delay from 2 seconds to 3 seconds. | Input data (in this case 2023 and 3) | Text field changes from 2017 to 2023 and delay from 2 to 3 | ✔ |
| 2. | User clicks on the ‘Simulate’ button. | Previously loaded csv file with information. | System executes the simulation and indicates the changes of the population on the map itself and the demographics of the population in the respective data charts. | ✔ |

### TEST CASE 4

Test case description: Saving city to database

Preconditions: Program is running

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User clicks on the button to add a city. |  | System shows the form for adding a city. | ✖  Not implemented as of iteration one |
| 2. | User inputs values such as population, city name, coordinates, birth and death rate, immigration and emigration rates and clicks on button to save city. | Input data | System validates data for correct format then saves data to the database. System shows data to user. | ✖ |
| 3. | User compares shown data with input data. |  | Shown data matches with input. | ✖ |
| 4. | User chooses to export city to file. | Data from database | System generates csv file with city data. | ✖ |
| 5. | User compares data in file with input data. |  | Data in file matches input. | ✖ |

### TEST CASE 5

Test case description: Loading city data from database.

Preconditions: Program is running.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User clicks on the button to load a city from database. | Cities in database | System shows a list of all available cities. | ✖  Not implemented as of iteration one |
| 2. | User chooses a city to load by clicking on load city button. | Data for chosen city from database | System loads data from the database to the screen for the selected city. | ✖ |
| 3. | User compares shown data with data from database. |  | Shown data matches data stored in database. | ✖ |

### 

### TEST CASE 6

Test case description: Saving city to a file

Preconditions: Program is running.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User clicks on the button to save a city. | City data in program | System shows a dialog with directory options. | ✔  Directory dialog is shown. |
| 2. | User chooses a directory to save the file in. |  | System saves data for city to file. | ✔  File is created and data for city is exported. |
| 3. | User compares data in file with data in program. | File data and data in program | Data in file matches city data in program. | ✔  Data in file matches with data in program. |

## NON-FUNCTIONAL REQUIREMENTS

### TEST CASE 7

Test case description: Starting program in a certain time.

Preconditions: Simulation program is present on user computer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User starts executable for simulation program. |  | Program starts and is ready for use within 10 seconds. System shows main form on screen. | ✔  Program loads in specified time and shows sample data. |

### TEST CASE 8

Test case description: Loading data from a file in a certain time.

Preconditions: Program is running and files are provided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Step description** | **Test data** | **Expected result** | **Actual result** |
| 1. | User clicks on the import a file button. |  | System shows a choose file dialog. | ✔  File dialog is shown. |
| 2. | User chooses a file to import. | Available city files (csv). | System loads data from the file within 5 seconds to the screen. | ✔  Data is loaded and charts are updated in the specified time. |
| 3. | User compares shown data with data from chosen file. |  | Shown data matches data from chosen file. | ✔  Data in charts matches data in file. |