Master Degree in
Computer Science

CADOCS
Conversational Agent for the Detection of Community Smells

Pre-Maintenance Test Case Specification

https://github.com/Nuri22/csDetector

Team Members

Gianmario Voria
g.voria6@studenti.unisa.it

Antonio Della Porta
a.dellaporta26@studenti.unisa.it

Reviewer

Stefano Lambiase
slambiase@unisa.it

18th July 2022

Computer Science Department
University of Salerno

Contents

1 Introduction

In this document, we will discuss the testing activities available for the CSDETECTOR tool before our maintenance and evolution process. It is important to note that Almarimi et. al. did not provide any test case.

After the reverse engineering performed on the tool, we found ourselves with the **functional requirements** shown in the Table 1

Table 1 CSDETECTOR Functional Requirements

FR_1: Community Smell Detection	The system must be able to detect and show community smells on a given GitHub repository.
FR_2: GitHub Repository Management	The system must be able to download a specified GitHub repository and access its content.
FR_3: Project's Metrics Computation	The system must be able to gather metrics of the given software project by analyzing both the software and the sentiment of the messages of commits, issues and pull requests.
FR_4: Persistence of the metrics	The system must be able to locally write and read metrics of the project in account.

We will take in account the testing of the FR_1, so we will perform **functional testing** in the context of testing the whole system.

2 Functional Testing of FR_1

The tool CSDETECTOR is able to detect community smells on a given repository. The execution of this tool relies on command line input, which for instance are:

- -p: GitHub PAT—Personal Access Token—used to query the repository;
- -r: Repository URL;
- -s: Directory containing Sentistrength, a library used for sentiment analysis;
- -o: Directory that will contain the output;
- -sd(Optional): Starting date from which commit have to be analyzed.

For this reason, a possible test case could be a *quintuplet* of strings, each representing a single parameter. Since we found ourselves with no existing test cases, we will create a set of input for the tool using the **Category Partition** technique.

Also, being the functional requirement the identification of community smells in a repository among a set of ten possible smells, we will have to ensure us that each of the single smell can be detected.

2.1 Test Frame

Description	
The system must be able to be integrated and executed through an Adapter	
Parameters	
gitRepository, *gitPAT*, *sentiFolder*, *outputFolder*	
Categories	
Category Name	**Category Value**
gitRepository	• GR1: gitRepository is valid • GR2: gitRepository is present but not represent a valid GitHub repository (the string passed is not a valid URL or a malformed one) • GR3: gitRepository is not valid (passed as None type)
gitPAT	• GP1: gitPAT is present and valid • GP2: gitPAT is not present (passed as None type)
sentiFolder	• SF1: sentiFolder is present and contains the SentiStrength classes and data • SF2: sentiFolder is present but not contains the SentiStrength classes and data • SF3: sentiFolder is not present (passed as None type)
outputFolder	• OF1: outputFolder is present • OF2: outputFolder is present but not represent a valid folder in the file system • OF3: outputFolder is not present (passed as None type)
Costraints	
Attribute	**Costraint**

gitRepository	• GR1: gitRepository is valid **[property gitRepository]** • GR2: gitRepository is present but not represent a valid GitHub repository (the string passed is not a valid URL or a malformed one) **[error]** • GR3: gitRepository is not valid (passed as None type) **[error]**
gitPAT	• GP1: gitPAT is present and valid **[if gitRepository] [property gitPAT]** • GP2: gitPAT is not present (passed as None type) **[error]**
sentiFolder	• SF1: sentiFolder is present and contains the SentiStrength classes and data **[if gitPAT] [property sentiFolder]** • SF2: sentiFolder is present but not contains the SentiStrength classes and data **[error]** • SF3: sentiFolder is not present (passed as None type) **[error]**
outputFolder	• OF1: outputFolder is present and represent a valid location **[if gitPAT] [property outputFolder]** • OF2: outputFolder is present but not represent a valid folder in the file system **[error]** • OF3: outputFolder is not present (passed as None type) **[error]**

Table 4 Test Frame for csDetector's input.

Test Case ID	Test Frame	Result
TC-FR.1_1	GR1, GP1, SF1, OF1	Success: the execution of the tool succeeds and the community smells detected are shown in console
TC-FR.1_2	GR1, GP1, SF1, OF2	Error: the system warns the user that the folder specified does not exist
TC-FR.1_3	GR1, GP1, SF1, OF3	Error: the system inform the user that an output folder is needed
TC-FR.1_4	GR1, GP1, SF2, OF1	Error: the system warns the user that the senti folder indicated does not contain the needed files
TC-FR.1_5	GR1, GP1, SF3, OF1	Error: the system inform the user that the folder with the SentiStrength files is needed
TC-FR.1_6	GR1, GP2, SF1, OF1	Error: the system inform the user that a GitHub PAT is needed to clone the repository
TC-FR.1_7	GR2, GP1, SF1, OF1	Error: the system warns the user that the URL is invalid or malformed
TC-FR.1_8	GR3, GP1, SF1, OF2	Error: the system inform the user that a GitHub repository URL is needed to start the tool

2.2 Test Case Specification

Table 5 Test Case TC-1_1

Test Case ID	**Test Frame**
TC-FR-1_1	**GR1, GP1, SF1, OF1**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitRepository*: https://github.com/tensorflow/serving • *gitPAT*: the GitHub PAT created • *sentiFolder*: senti • *outputFolder*: out	
Oracle	
The tool prints as output a list of the Community Smells detected.	

Table 7 Test Case TC-FR-1_2

Test Case ID	**Test Frame**
TC-FR-1_2	**GR1, GP1, SF1, OF2**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitRepository*: https://github.com/tensorflow/serving • *gitPAT*: the GitHub PAT created • *sentiFolder*: senti • *outputFolder*: output	
Oracle	
The tool is not able to save output information and stops its execution without detecting smells.	

Table 9 Test Case TC-FR-1_3

Test Case ID	**Test Frame**
TC-FR-1_3	**GR1, GP1, SF1, OF3**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitRepository*: https://github.com/tensorflow/serving • *gitPAT*: the GitHub PAT created • *sentiFolder*: senti	
Oracle	
The tool warns the user that a needed parameter is missing and stops its execution without detecting smells.	

Table 11 Test Case TC-FR-1_4

Test Case ID	**Test Frame**
TC-FR-1_4	**GR1, GP1, SF2, OF1**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitRepository*: https://github.com/tensorflow/serving • *gitPAT*: the GitHub PAT created • *sentiFolder*: sentifolder • *sentiFolder*: out	
Oracle	
The tool does not find the required modules in the folder specified and stops its execution without detecting smells.	

Table 13 Test Case TC-FR-1_5

Test Case ID	**Test Frame**
TC-FR-1_5	**GR1, GP1, SF3, OF1**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitRepository*: https://github.com/tensorflow/serving • *gitPAT*: the GitHub PAT created • *sentiFolder*: out	
Oracle	
The tool warns the user that a needed parameter is missing and stops its execution without detecting smells.	

Table 15 Test Case TC-FR-1_6

Test Case ID	**Test Frame**
TC-FR-1_6	**GR1, GP2, SF1, OF1**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitRepository*: https://github.com/tensorflow/serving • *sentiFolder*: senti • *sentiFolder*: out	
Oracle	
The tool warns the user that a needed parameter is missing and stops its execution without detecting smells.	

Table 17 Test Case TC-FR-1_7

Test Case ID	**Test Frame**
TC-FR-1_7	**GR2, GP1, SF1, OF1**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitRepository*: invalidurl • *gitPAT*: the GitHub PAT created • *sentiFolder*: senti • *sentiFolder*: out	
Oracle	
The tool can't find a repository at the specified URL and stops its execution without detecting smells.	

Table 19 Test Case TC-FR-1_8

Test Case ID	**Test Frame**
TC-FR-1_8	**GR3, GP1, SF1, OF1**
Pre-conditions	
• the tool's environment has been installed and so the tool is executable; • a folder named **"senti"** has been created and contains SentiStrength data; • a folder named **"out"** has been created; • a **GitHub PAT** has been created and it correctly works for API calls.	
Flow of events	
The user writes the following parameters as input of the tool: • *gitPAT*: the GitHub PAT created • *sentiFolder*: senti • *sentiFolder*: out	
Oracle	
The tool warns the user that a needed parameter is missing and stops its execution without detecting smells.	