“Telesales GUI”

Functional Specifications Document

Rashad Asgarbayli

January 9, 2015

Contents

1 Introduction

We already have a batch tool named Telesales, that starts and runs automatic tests. But to configure the tool to run and do its work correctly, is more complexer. Because a co-worker needs minimum XML knowledge to create a right XML-Configuration file, and he/she has to do this manually. We need an application part, a GUI to make our configuration file faster and correct.

Introducing a new Java(FX)-based GUI tool - "Telesales GUI" for our batch application. "Telesales GUI" will help You to create XML-Configuration files in an instant, thanks to its user-friendly interface and XML-creator engine. The co-worker only needs to enter some core data that needed for the batch-application. "Telesales GUI" will than analyze the inputted data in the fields and generate a matching XML-Configuration files.

2 Objectives

"Telesales GUI" will be used by co-workers of QA, to create/generate XML-Configuration files for the Telesales faster and easy.

2.1 Mandatory Criteria

- Create XML-Configuration files for Telesales
- Open and edit XML-Configuration files created in "Telesales GUI".

2.2 Facultative Criteria

- "Telesales GUI" starts Telesales batch application by its own using current configuration just clicking "Start"-Button.

2.3 Excluded Criteria

- "Telesales GUI" can not create two or more XML-Configuration files at the same time.
- "Telesales GUI" alone can not run tests on its own, it only generates required XML-files or can start the Telesales batch application just passing right parameters.

3 Product Usage

"Telesales GUI" will be used to create and edit XML-Configuration files for the Telesales batch application.

3.1 Users

Users of the application are co-workers of the QA

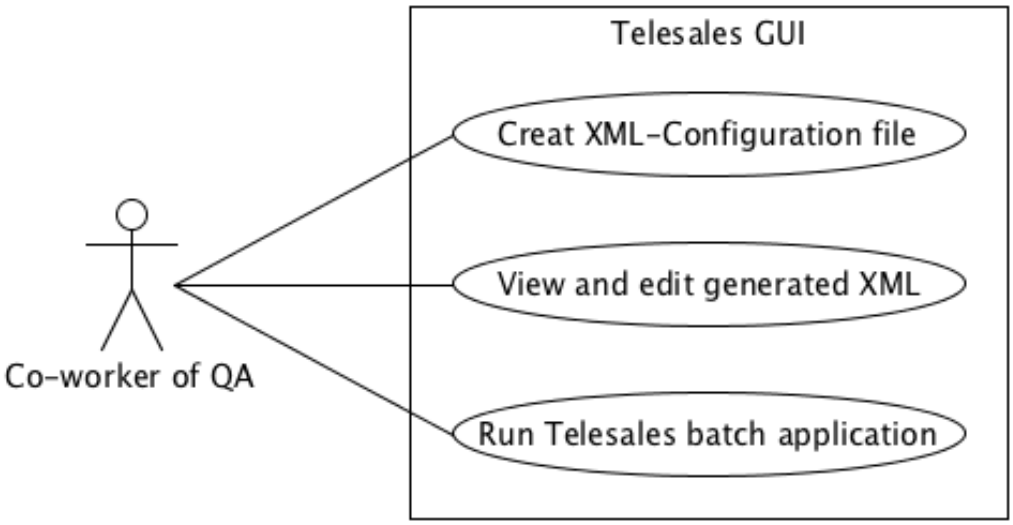

Figure 1: Product usage

4 Product Environment

4.1 Environment for "Telesales GUI"

"Telesales GUI" is a Java (JavaFX) based application. Because of the java nature, it will run on all major OS's (Windows, Mac OS X, Linux etc.) as long as Java Runtime libraries installed. The minimum required Java runtime environment is v1.8 (v8) or higher.

4.2 Hardware

Minimum System Requirements depends of the minimum system requirements of Java and JavaFX.

5 Functional Requirements

5.1 Mandatory Functional Requirements

/FM010/ Create XML-Configuration file
User is able to create/generate XML-Configuration file.

/FM020/ Open and view XML-Configuration file
User is able to open XML files, if the file created or generated using "Telesales GUI".

/FM030/ Save XML-Configuration file
User is able to edit and save the existed XML-Configuration files using **FM020** and **FM030** combined and just save the new XML-Configuration files using **FM010** and **FM030** combined.

5.2 Facultative Functional Requirements

/FF010/ Run Telesales batch
User is able to start Telesales batch application just clicking "Start"-button. "Telesales GUI" runs the batch application for the current opened or created configuration passing the right parameters into the terminal.

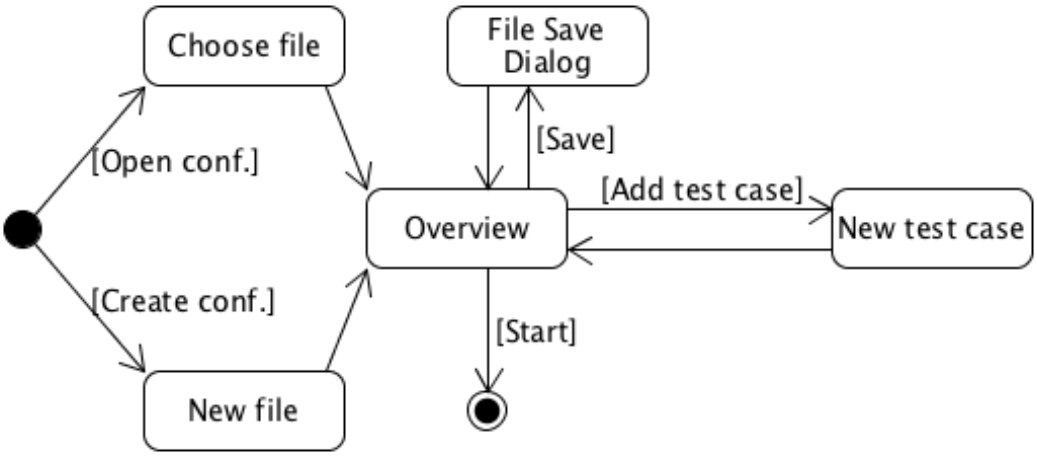

Figure 2: Activity diagram of the functional requirements

6 Product Data

Each time "Telesales GUI" generates two XML-Configuration files. One for test configuration and another one for the tester's username and password. The "-batch" (for configuration file) and "-login-data" (for testers login information) endings will be added automatically into the names of the files.

6.1 Configuration data

/D010/ country
Tester have to choose a country from the list of countries for testing. The chosen one will be added automatically into the all needed places in the XML-File.

/D020/ URL
Tester have to enter the URL for testing. The entered URL will be added automatically into the all needed places in the XML-File.

/D030/ customer-number
It is a number of a test customer account and necessary for testing.

/D040/ tariff
Tester have to enter the required tariff information for testing. The entered information will be added automatically into the all needed places in the XML-File.

/D050/ tariff-addons
Tester can enter the additional tariff options for testing (optional). The entered data will be added automatically into the all needed places in the XML-File.

/D060/ domain
Tester have to enter the domain for testing. The format looks like "yourdomain.de". A time stamp will be added automatically before the last dot. The entered data will be edited in background automatically and automatically into the needed place in form "yourdomain{TIMESTAMP}.de" in the XML-File.

/D070/ domain-bundle
Tester can set the domain-bundle option to the true or false (default is false) for testing (optional). The data will be added automatically into the needed place in the XML-File.

Example of a generated XML-Configuration file:

```
<?xml version="1.0"?>
<telesales-buying-agent-batch>
  <telesales-urls>
   <telesales-url country="DE">https://hosting-ts-de.ts-host.gem1.sales.united.domain:9613/</telesales-url>
   <telesales-url country="ES">https://hosting-ts-es.ts-host.gem1.sales.united.domain:9613/</telesales-url>
   <telesales-url country="FR">https://hosting-ts-fr.ts-host.gem1.sales.united.domain:9613/</telesales-url>
    …
  </telesales-urls>
  <customers>
    <customer country="DE">
      <customer-number>7156019</customer-number>
    </customer>
    <customer country="FR">
      <customer-number>207958780</customer-number>
    </customer>
    …
  </customers>
  <orders>
    <order>
      <country>DE</country>
      <tariff>tariff-basic</tariff>
      <tariff-campaign-control>tariff-toggle</tariff-campaign-control>
      <tariff-addons>
        <tariff-addon id="presales.articles.slot-eshop-addon">opt-addon-eshop-basic</tariff-addon>
      <tariff-addon id="presales.articles.slot-sitelock-basic-addon">opt-addon-sitelock-basic</tariff-addon>
        <tariff-addon id="presales.articles.slot-seotool-addon">opt-addon-seotool</tariff-addon>
        …
      </tariff-addons>
      <domain>deinedomain{TIMESTAMP}.de</domain> <!-- {TIMESTAMP} will be replaced by yyyyMMddHHmmss -->
```

```
      <domain-bundle>false</domain-bundle>
    </order>
    <order>
      <country>FR</country>
      <tariff>tariff-basic</tariff>
      <tariff-campaign-control>tariff-toggle</tariff-campaign-control>
      <tariff-addons />
      <domain>votredomain{TIMESTAMP}.fr</domain> <!-- {TIMESTAMP} will be replaced by yyyyMMddHHmmss -->
      <domain-bundle>true</domain-bundle>
    </order>
  </orders>
</telesales-buying-agent-batch>
```

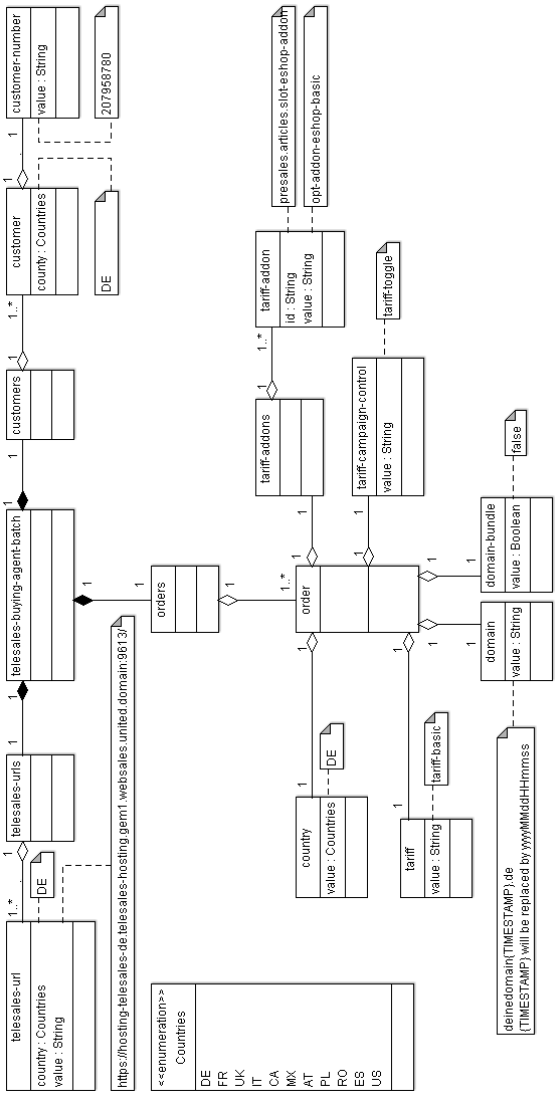

Figure 3: UML Class diagramm of the Configuration file

6.2 Login data

This data will be used as a login data of the testers.

/DC010/ username
Tester's username.

/DC020/ password
Tester's password.

Example of a generated XML-Login-data file:

```
<?xml version="1.0"?>
<telesales-buying-agent-login-data>
    <username>DeinBenutzername</username>
    <password>DeinPasswort</password>
</telesales-buying-agent-login-data>
```

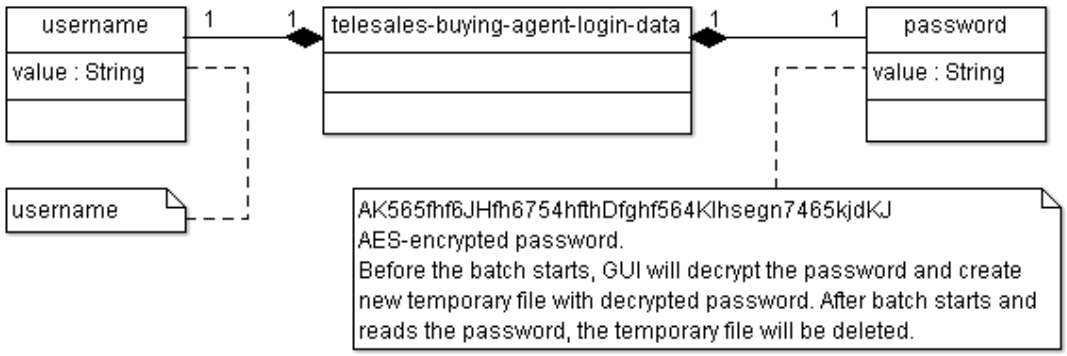

Figure 4: UML Class diagramm of the Login data file

7 Nonfunctional Requirements

/NF010/ Bigger GUI Elements
Bigger GUI elements provides better readability and makes it easy to work on touchscreen devices.

8 Global Test Cases and Scenarios

In this section I describe the tests for the functional requirements in steps and with the wanted outcome. (*Not ready yet!*)

8.1 Basic features

/TB010/ TestName
Description.

8.2 Optional features

/TO010/ TestName
Description.

9 System Model

Figure 5: System Model

The main model applied to the product is the model-view-controller (MVC model).

Blablabla

9.1 Graphical User Interface

Blablabla

Figure 6: GUI Blablabla

10 Glossary

GUI
: Graphical User Interface

QA
: Quality Assurance

User
: Co-worker of the QA