

Software Requirements Specification

ETL Support 2019 - Group 2

January 7, 2019

Team Members

- Elia Vicentini (Scrum Master)
- Simone Gandini
- Luca Landolfo
- Enrico Marocchio
- Adrian Munteanu

Document Storage

This document is stored in a GitHub [repository](#), where you can also find all the other types of documentation regarding this project.

Document Owner

Enrico Marocchio is responsible for the development and maintenance of this document.

Table of Contents

1 Introduction

- 1.1 Overview
- 1.2 Goals and Objectives
- 1.3 Scope

2 General design constraints

2.1 User Characteristics

2.2 Mandated Constraints

3 NonFunctional Requirements

3.1 Operational Requirements

3.2 Performance Requirements

3.3 Documentation and Training

3.4 Interface

4 Functional Requirements

4.1 Required Features

4.2 Optional Features

Revision History

Version	Date	Name	Description
1	31/12/2019	Marocchio Enrico	Initial Document
2	2/01/2020	Marocchio Enrico	General design and NonFunctional requirements
3	3/01/2020	Marocchio Enrico	Functional Requirements and review

1) Introduction

1.1) Overview

The ETL-Support program is a python script that will allow to backup data from the MarconiTT team programm managing and updating it into a server called (SIGMAQ). Then this backup can be used via Reporting Service of the school providing informations about the school timetable

1.2) Goals and Objectives

The main goal of this project is to modify and make the output of marconiTT program readable and use it with electronic circuit like (SOM) that displays a class timetable next to a room

1.3) Scope

Backing up school timetable and making it readable via Reporting Services

2) General design constraints

2.1) User Characteristics

The program will be used most of the time automatically by a machine but it can be used manually from an user via command prompt launching the program in this way it will print at screen all the data modified

2.2) Mandated Constraints

The application will run in Windows platform. Better if in a virtualized environment. This platform was used for his compatibility with python 3.7 that is mandatory for this program to work

3) NonFunctional Requirements

3.1) Operational Requirements

Usability: Reading the user manual is not required to be able to use the application because it'll run by itself but for personal use and manual run it's recommended to read it

3.2) Performance Requirements

Maintainability: Changes made to the Program can change the output and can cause errors in output or input formats

3.3) Documentation and Training

Documentation can be found in the GitHub site of the Scrum Master and no training is needed in order to use the application

3.4) Interface

In the application the interface is Command line (CLI) and it work automatically at the start so there is no interaction between user and application

4) Functional Requirements

4.1) Optional Features

The purpose of this application is to work automatically but if an user wants to run the program in live in order to do that he need to enter the folder where it is stored via CMD and run it from here with the option “python” before.