TES

0.1v

TEMPLATE EXECUTION SOFTWARE(TES)

The purpose of this document is to provide with a template for documenting TES.

**Document Control:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | | |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| 19/11/2022 | 0.1v | Group03 | Initial Draft | | | |  | |
| 25/11/2022 | 0.2v | Group03 | Added Flowchart | | | |  | |

Table of Contents

[1 Introduction 5](#_Toc120442410)

[1.1. Acronyms/Abbreviations 5](#_Toc120442411)

[1.2. Project Purpose 5](#_Toc120442412)

[1.3. Key Project Objectives 5](#_Toc120442414)

[⮚ Have to put place holder variable in template. 5](#_Toc120442415)

[⮚ Atleast one field of candidate should be uniq. 5](#_Toc120442416)

[⮚ Project Scope and Limitation 5](#_Toc120442417)

[1.7 Assumptions, Dependencies & Constraints 6](#_Toc120442418)

[● Organization has machine capable of running a UNIX based operating system. 6](#_Toc120442419)

[● C source code can be compiled on the machine. 6](#_Toc120442420)

[● The user has some storage space to store the data. 6](#_Toc120442421)

[1.8 Risks 6](#_Toc120442422)

[2 Design Overview 6](#_Toc120442423)

[2.1 Design Objectives 6](#_Toc120442424)

[2.2.1 Design Alternative 6](#_Toc120442425)

[2.2.2 Reuse of Existing Common Services/Utilities 6](#_Toc120442426)

[2.2.3 Creation of New Common Services/Utilities 6](#_Toc120442427)

[2.2.4 User Interface Paradigms 6](#_Toc120442428)

[2.2.5 System Interface Paradigms 7](#_Toc120442429)

[● 64bit Machine capable of running UNIX based operating system. 7](#_Toc120442430)

[● Storage space to store the data. 7](#_Toc120442431)

[2.2.6 Error Detection 7](#_Toc120442432)

[2.2.7 Memory Management 7](#_Toc120442433)

[2.2.8 Performance 7](#_Toc120442434)

[2.2.9 Security 7](#_Toc120442435)

[Some of the factors that are identified to protect the software from accidental or malicious access, use modification. Keep specific log or history data sets. 7](#_Toc120442436)

[2.2.10 Concurrency and Synchronization 7](#_Toc120442437)

[2.2.11 Housekeeping and Maintenance 7](#_Toc120442438)

[3.4 System Interfaces 8](#_Toc120442439)

[3.4.1 Internal Interfaces 8](#_Toc120442440)

[3.4.2 External Interfaces 8](#_Toc120442441)

[9](#_Toc120442442)

[9](#_Toc120442443)

[9](#_Toc120442444)

[9](#_Toc120442445)

[9](#_Toc120442446)

[9](#_Toc120442447)

[9](#_Toc120442448)

[9](#_Toc120442449)

[9](#_Toc120442450)

[5 Environment Description 10](#_Toc120442451)

[5.1Time Zone Support 10](#_Toc120442452)

[5.2 Language Support 10](#_Toc120442453)

[5.4 Server-Side Requirements 10](#_Toc120442454)

[5.4.1 Deployment Considerations 10](#_Toc120442455)

[5.4.2 Application Server Disk Space 10](#_Toc120442456)

[5.4.3 Database Server Disk Space 10](#_Toc120442457)

[5.4.4 Integration Requirements 11](#_Toc120442458)

[5.4.6 Network 11](#_Toc120442459)

[5.4.7 Others 11](#_Toc120442460)

[5.5 Configuration 11](#_Toc120442461)

[5.5.1 Operating System 11](#_Toc120442462)

[5.5.2 Database 11](#_Toc120442463)

[5.5.3 Network 11](#_Toc120442464)

[5.5.4 Desktop 11](#_Toc120442465)

[6 References 11](#_Toc120442466)

### 1 Introduction

The ‘Template Execution Software’ project will develop to overcome the time-consuming problem of manual system. The ‘TEMPLATE EXECUTION SOFTWARE’ project will be developed to overcome the time consuming problem of manual system. Here we collect the one template file and multiple data files here we put the placeholder variable in the template file. Here we replace the variable with value from the data file. The project will reduce the manual process in managing examinations and all issues regarding that. The users which are use this system don’t need to high computing knowledge and also system will inform them while entering invalid data.

## Acronyms/Abbreviations

|  |  |
| --- | --- |
| QES | Quiz Evaluation Software |
| CSV | Comma Separated values |

## Project Purpose

# Template Execution software using C and its various supporting toolsThe ‘TEMPLATE EXECUTION SOFTWARE’ project will developed to overcome the time consuming problem of manual system. Here we collect the one template file and multiple data files here we put the placeholder variable in the template file. Here we replace the variable with value from the data file. The project will reduce the manual process in managing examinations and all issues regarding that. This project helps the examiners to manage their services in a good way and provide a better service to their users.

## Key Project Objectives

## Have to put place holder variable in template.

## Atleast one field of candidate should be uniq.

## Project Scope and Limitation

**1.5.1 In Scope**

To replace the variable with value which is stored in datafiles.

**1.5.2 Out Scope**

It is not a real-time project.

**1.6 Functional Overview**

 1.replace:replace the old word with new word

2.load template: Here we load the template files.

3load data : here we load the data files.

4.process replacements: here the process is replacing from one data file to another.

## 1.7 Assumptions, Dependencies & Constraints

## Organization has machine capable of running a UNIX based operating system.

## C source code can be compiled on the machine.

## The user has some storage space to store the data.

## 1.8 Risks

When we enter wrong data to particular candidate it becomes complex

### 2 Design Overview

Operating Environment for the Template Execution Software is as follows:

* Operating System: Any UNIX Based OS
* Compiler: GCC or similar to compile source code written in C programming language.

## 2.1 Design Objectives

* Here we have to put all the data files into data csv. which helps less usage of memory. Less time consumption.

### 2.2.1 Design Alternative

It explains the connection between template file and datafiles.

### 2.2.2 Reuse of Existing Common Services/Utilities

The project does not reuse any new common services/utilities.

### 2.2.3 Creation of New Common Services/Utilities

The project does not create or use any new common services/utilities.

### 2.2.4 User Interface Paradigms

1. GUI: The application does not use Graphical User Interface.
2. CLI: The application uses Command Line Interface to accept console commands by users and perform the needful functions.

### 2.2.5 System Interface Paradigms

### 64bit Machine capable of running UNIX based operating system.

### Storage space to store the data.

### 2.2.6 Error Detection

if the file is empty then we get the error here.

### 2.2.7 Memory Management

Here we put the all data files into one data csv soo here we can save the memory space. And time too.

### 2.2.8 Performance

The Application is developed to run through CLI on UNIX based system as the machine can run the operating system along with necessary dependencies without any flaws there are no additional requirements.

### 2.2.9 Security

# Some of the factors that are identified to protect the software from accidental or malicious access, use modification. Keep specific log or history data sets.

### 2.2.10 Concurrency and Synchronization

NA

### 2.2.11 Housekeeping and Maintenance

Clearing the memory buffer from the system.

## 3.4 System Interfaces

### 3.4.1 Internal Interfaces

GUI: The application does not use Graphical User Interface.

CLI: The application uses Command Line Interface to accept console commands by users and perform the needful functions

### 3.4.2 External Interfaces

Hardware Requirements are as follows:

* 64bit Machine capable of running UNIX based operating system.
* Storage space to store the data

**4**.**DIAGRAMETIC REPRESENTATION:**

### 

### 

Here we have to create template file

And multiple data files in data.csv

Template Execution Software

### 

### 

Load the multiple datafiles here we storing variable and value pairs

Load the template file here we placing the placeholder variables

### 

### 

We have replace the variable with values. Here old word is replaced with new word

### 

### 

Here we can Extract the tokens

This helps to complete the process in less time

### 

### 

Process replacement function which helps to continue the process in the loop manner .it is because we using multiple data types

### 5 Environment Description

### 5.1Time Zone Support

It will support time zone as per Indian standard time (IST) in (GMT +5:30) and UST standard.

## Language Support

C language in vi editor and compilation using make file and GCC. The Linux commands are used to do that task with specified commands.

**5.3 User Desktop Requirements**

User desktop requires a Linux environment, Operating system of Linux Debian or Terminal x86\_64 GNU/Linux and kernel version 4.4.0-19041-Microsoft #1237Microsoft and reliable internet connectivity.

## 5.4 Server-Side Requirements

In server side,

● Disk space – Minimum 150GB

● Uninterrupted connectivity 24x7

● Monitor long running jobs, to reduce the server load

### 5.4.1 Deployment Considerations

Deployment considerations are,

● 500Mhz Processor

● 120GB HDD CPU

● minimum 4GB RAM

● Network connectivity

### 5.4.2 Application Server Disk Space

Disk space -Minimum 150GB

### 5.4.3 Database Server Disk Space

NA

### 5.4.4 Integration Requirements

Project integration management involves coordinating all the project elements.

### 5.4.6 Network

NA

### 5.4.7 Others

NA

## 5.5 Configuration

### 5.5.1 Operating System

* Operating system –Linux.
* RAM - 4GB or more.
* Processor - i3/i5.

### 5.5.2 Database

NA

### 5.5.3 Network

NA

### 5.5.4 Desktop

Minimum Windows 10, 8gb Ram with i5 configuration is required.

### 6 References

* [www.stackoverflow.com](http://www.stackoverflow.com)
* <https://www.javatpoint.com/file-handling-in-c> for file handling concept
* <https://www.javatpoint.com/linux-commands> Linux Command

**Change Log**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TES Template Version Control** | | | | | |
|  |  |  |  |  |  |
| **Date** | **Version** | **Author** | | **Description** | |
| 18-Nov-2022 | 1.0 | Group 03 | | Initial Version | |