

Software Architecture Document (SAD)

Team Slprogramming

Editeur de texte (TextEd)

KONDOMBO ISMAEL BHALE KOUANGBE SILVER AFRAITANE AYAL KONAN ABRAHAM KAMATE ISSIAKA

29 MARS 2021

Chapitre 1 : INTRODUCTION

1° OBJET

Ce document de conception logiciel (SDD) est mis sur pied pour décrire l'architecture et la structure du système logiciel TextEd ainsi que l'implémentation des exigences spécifiques retenues dans le document de spécification (SRS). Nous prions toutes personnes de d'abord prendre note des exigences spécifiques dans le document de spécification avant de lire ce document qui sera une implémentation détaillée des dits exigences.

2° OBJECTIF

L'établissement de ce document de conception a plusieurs objectifs qui sont les suivants :

- Décrire la structure fonctionnelle du système logiciel ainsi que les données et l'algorithme implémenté
- Juger du changement des exigences spécifiques
- Vérifier l'implémentation effective des exigences spécifiques
- Faire les tests unitaires du système logiciel
- Aider à la maintenance du système logiciel

3°) CLIENT

Le client pour ce projet est :

Nom : Dr. DIALLO

Prénom : MOHAMED

Adresse :

Téléphone : 225 05 56 38 40 14

Email : diallo.med@gmail.com

4° EQUIPE

L'équipe sur ce projet est l'équipe Siprogramming constitué de :

| Membre de l'équipe | Numéro de téléphone |
|--------------------|---------------------|
| Kondombo Ismaël | 225 05 56 38 40 14 |
| Bhale Silvere | 225 |
| AFRAITANE Ayal | 225 |
| Konan Abraham | 225 |
| Kamate issiaka | 225 |

5° VUE GLOBALE DU DOCUMENT

CHAPITRE 2 : ARCHITECTURE DU SYSTÈME

L'architecture du système décrit en détail les fonctions, méthodes et objet utiliser lors de la création de notre application TextEd. Nous essayerons de fournir des détails sur l'algorithme utiliser et le processus de codage.

1° MODELE DE REFERENCE

Le modèle ci-dessous présente la manière dont sera détaillé les objets et méthodes du logiciel TextEd.

BUT : Décrit la fonction ou l'objet

PROTOTYPE : Présente comment l'objet ou la méthode est appelée

ENTRE : Décrit les paramètres d'entrés

SORTIE : Description des valeurs retournés

RESTRICTIONS : Description de toutes les restrictions en entré

APPELER PAR : objet ou fonction qui appelle notre objet ou méthode

APPEL A : Objet ou fonction que notre objet ou fonction appelle

ALGORITHME : Description de l'algorithme ou la procédure

2° INTERFACE GRAPHIQUE (TKinter)

BUT : bibliothèque de python qui nous offre un ensemble outil pour créer nos différents widgets

PROTOTYPE : tkinter()

ENTRE : non définie

SORTIE : no definie

APPELER PAR : class_editeur.py

APPEL A : Tk()

ALGORITHME :

- Création de la fenêtre principale
- Ajout de la zone de texte
- Ajout des bars de défilement
- Ajout de menu

3° Tk()

BUT : initialisation de la fenêtre principal utilisateur

PROTOTYPE : tkinter.Tk()

ENTRE : non définie

SORTIE : non définie

RESTRICTIONS : Les restriction imposées par l'interface graphique de python

APPELER PAR : class_editeur.py

APPEL A : Toute les méthodes nécessaires à Tk()

ALGORITHME : Automatiquement généré par tkinter

4° Menu

BUT : Permet d'afficher les différentes fenêtres

PROTOTYPE : Tk.Menu()

ENTRE : non définie

SORTIE : non définie

RESTRICTIONS : Les restriction définie par l'objet Tk

APPELER PAR : fenêtre_principal

APPEL A : Rien

ALGORITHME : Automatiquement générer par Tk.Menu()

5° class_editeur

BUT : définition de toutes les méthodes qui permettent de rendre dynamique notre fenêtre

PROTOTYPE : class_editeur est appelé dans TextEd en tant que module

ENTRE : non définie

SORTIE : non définie

RESTRICTIONS : Les restrictions sur les appels de module en python

APPELER PAR : TextEd.py

APPEL A : Rien

ALGORITHME : classe d'objet

VERSION 2

Pourquoi la version 2 ?

La version 2 de TextEd vient tenter de répondre à nos objectifs qui sont données un éditeur tout fonctionnel, mais aussi offrir une interface plus moderne, ce que ne peut fournir les modules natifs de python n'offre pas.

La version 2 de l'Editeur de Texte peut se décomposer en deux parties.

Une partie **Front-End**, et une partie **Back-End**

-Le **Back-End** se compose tout uniquement de code écrit en **Python**

-le **Front-End** (Interface graphique) est fait en **React js**

Comment est fait le Front-End ?

Le Front-End n'est pas assez compliqué en matière de logique, on avait opté pour une clean architecture, mais après du au temps qui nous restait pour faire cette version, nous avons dû simplifier certains processus histoire de rester dans le temps imparti.

Just une connaissance en React js, en JavaScript et un peu de Typescript permet de savoir ce que faire le code qui se trouve sur le repository.

Comment est fait le Back-End ?

Le Back-End quant à lui est écrit en python et utilise un module appelé Eel.

Eel, nous permet de créer un **Webview** afin d'injecter à l'intérieur du code **html, css, et JavaScript**.

Un Webview est simplement un navigateur, mais plus précisément une fenêtre d'un navigateur. Ce qui est idéal et nous permet alors d'embarquer alors toutes les technologies utilisées dans le web, tels que html, css, js, vue.js, React js, sass, ainsi que les **Websockets**.

Ceci étant alors comment se fait alors la communication entre un langage telle que python et d'autre langage telle que JavaScript ?

Simplement Eel, nous permet d'exposer des fonctions python dans le code JavaScript et vice-versa (du code JavaScript disponible dans python).

Alors qu'est ce qui se passe en réalité avec ce Eel ?

Simplement Eel, crée un serveur et crée une connexion Websockets, sur lequel se fait la communication de python à JavaScript. Aussi **Eel** rend accessible nos fonctions python en exposant nos fonctions dans son API.

Mais pourquoi avoir utiliser python en Back-End et non pas alors JavaScript ?

Simplement JavaScript n'a pas la fonctionnalité de faire les manipulations telle qu'ouvrir un fichier, copier, créer un dossier sur la machine de l'utilisateur et etc. Pourtant **python** lui a cette faculté déjà embarquées.

Pour élargissement, si on voulait utiliser JavaScript, on aurait pu utiliser **node js** afin de nous fournir ces fonctionnalités d'accès système à JavaScript.

Bibliographie

[1] <http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf>

[2] Miles Izzo , November 6, 2002 , Software Design Document

[3] Pascal Ortiz ,23 février 2021, Documentation Tkinter

[4] <https://python.sdv.univ-paris-diderot.fr/>