

Générateur d'écran IHM

Cahier des charges

Table des matières

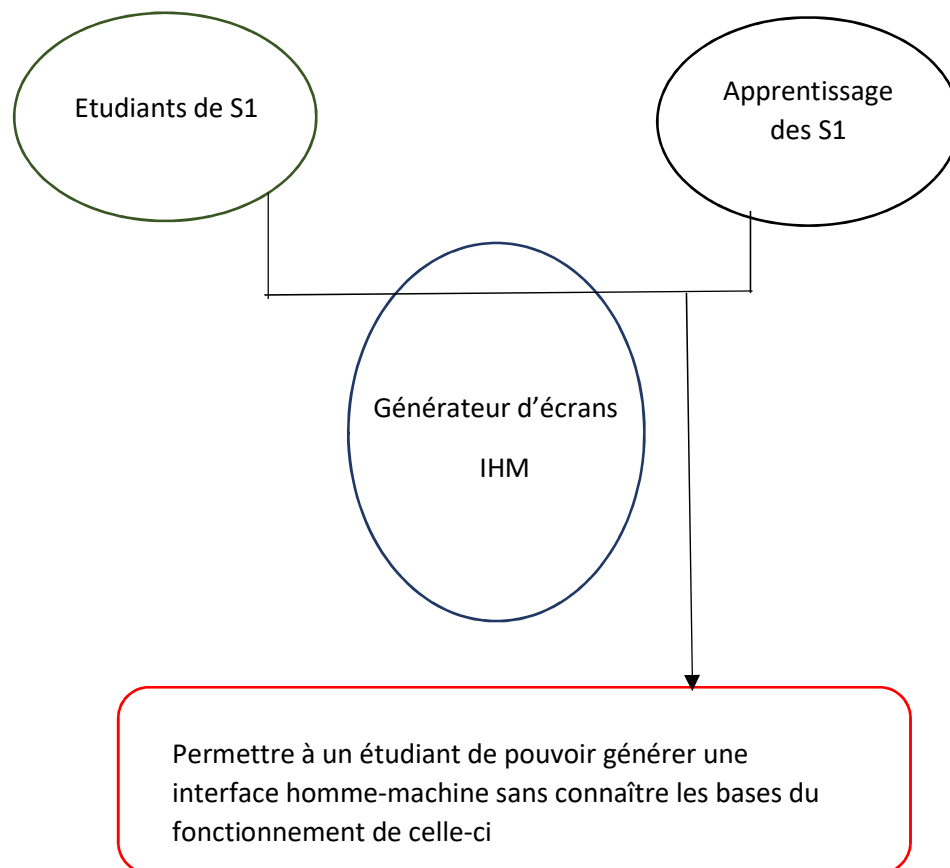
I. Présentation du projet	3
II. Enoncé du besoin	3
III. Validations du besoin	4
1. Pourquoi ce besoin existe-t-il ?	4
2. Ce besoin peut-il disparaître ?	4
3. Ce besoin peut-il évoluer ?	4
IV. Analyse fonctionnelle du besoin	4
1. Environnement du produit	4
2. Enoncé des fonctions de service	5
V. Appréciation des fonctions de service	5
VI. Hiérarchisation des fonctions de service	7

I. Présentation du projet

L'objectif de ce projet est de réaliser un logiciel générant des écrans sous forme d'interface homme-machine. Le produit sera utilisé dans le cadre de la formation de niveau post-bac (bac+1) en programmation JAVA. Il est principalement destiné aux étudiants débutants en programmation.

Le logiciel ainsi livré, devra être utilisé de manière simple par les étudiants qui devront intégrer quelques lignes de codes simples pour pouvoir visualiser une interface homme-machine. L'étudiant ses données devra écrire dans un fichier XML qui sera ensuite traité par le logiciel. Un manuel d'utilisateur sera livré avec le logiciel pour apprendre à l'utiliser.

II. Enoncé du besoin



III. Validations du besoin

1. Pourquoi ce besoin existe-t-il ?

Les étudiants de S1 n'ayant pas encore appris à manipuler des objets (en créer ou les utiliser de manière complexe) ne peuvent donc pas créer ni utiliser d'IHM graphique. C'est pourquoi cet utilitaire leur permettra de créer de manière simple (à travers une simple commande) une IHM graphique prenant la forme d'un formulaire et ensuite en extraire des données.

2. Ce besoin peut-il disparaître ?

Non, car les étudiants de S1 ont, chaque année, le besoin de créer des IHM et ainsi d'utiliser cet utilitaire.

3. Ce besoin peut-il évoluer ?

Bien sûr, même si le besoin principale restera le même (créer une IHM et en extraire des données), le format de celle-ci pourra évoluer avec l'évolution des classes java mises à disposition par les librairies (java.awt été javax.swing).

IV. Analyse fonctionnelle du besoin

1. Environnement du produit

Les acteurs :

Etudiants de S1 : personne qui créeront l'IHM graphique.

Utilisateur final : personne qui utilisera l'application créé par l'étudiant à l'aide de notre utilitaire

Ordinateur : machines sur lesquelles l'utilitaire sera présent

Contraintes :

C1 : Doit être simple d'utilisation

C2 : Doit être utilisable par un code JAVA

C3 : déclaration des composantes de l'IHM dans un fichier XML simplifié

2. Enoncé des fonctions de service

Fonction principale :

Permettre aux étudiants de S1 de créer facilement des IHM graphique sous forme de formulaire et d'en extraire les informations saisies.

Les différentes fonctionnalités attendues :

- Aspect Conception
 - C1 : Définir les composantes
 - C2 : Créer une nouvelle IHM
 - C3 : Extraire les données du formulaire
- Aspect Gestion des Erreurs
 - G1 : Afficher les erreurs de syntaxes
 - G2 : Afficher les erreurs de fichiers
- Aspect Paramétrage
 - P1 : Pouvoir changer la couleur de fond
 - P2 : Pouvoir changer la taille de la fenêtre
 - P3 : Pouvoir changer la taille de la police
- Aspect Ergonomie
 - E1 : Respecter les règles ergonomiques
 - E2 : Avertir en cas de surcharge visuelle
- Aspect Documentation
 - D1 : Documentation Utilisateur

V. Appréciation des fonctions de service

C1 : Définir les composantes

Comment : A travers l'écriture des différentes composantes dans un fichier XML simplifié

C2 : Créer une nouvelle IHM

But : Déclarer et instancier l'IHM à créer

Comment : A travers l'appel d'une méthode « static » qui créera pour l'étudiant l'IHM.

C3 : Extraire les données du formulaire

Comment : A travers des getters (getInt (nom : String), getString (nom : String), ...)

G1 : Afficher les erreurs de syntaxes

But : Bloquer l'exécution du programme et aider l'étudiant à corriger ses erreurs

Comment : A travers des boîtes de dialogue décrivant le plus précisément possible l'erreur

G2 : Afficher les erreurs de fichiers

But : identique à G1

Comment : identique à G1

P1 : Pouvoir changer la couleur de fond

But : Offrir à l'étudiant la possibilité de personnaliser son affichage

Comment : A travers l'appel à une méthode spécifique

P2 : Pouvoir changer la taille de la fenêtre

But : identique à P1

Comment : identique à P1

P3 : Pouvoir changer la taille de la police

But : identique à P1

Comment : identique à P1

E1 : Respecter les règles ergonomiques

But : créer une IHM lisible et facile à prendre en main

E2 : Avertir en cas de surcharge visuelle

But : Permettre à l'étudiant de savoir si son IHM sera lisible ou non

Comment : A travers un système d'avertissement au lancement de son programme

VI. Hiérarchisation des fonctions de service

Primordiales	C1 : Définir les composantes C2 : Créer une nouvelle IHM C3 : Extraire les données du formulaire
Importantes	D1 : Documentation utilisateur G2 : Afficher les erreurs de fichiers G1 : Afficher les erreurs de syntaxes
Non négligeables	E2 : Avertir en cas de surcharge visuelle E1 : Respecter les règles ergonomiques
Optionnelles	P1 : Pouvoir changer la couleur de fond P2 : Pouvoir changer la taille de la fenêtre P3 : Pouvoir changer la taille de la police