

# Projet : Simplexe

DUT Informatique 2<sup>ème</sup> année

Formation initiale

IUT de Vélizy

Université de Versailles-St-Quentin-en-  
Yveline

Étudiant(s) :

Claire BAUCHU, Louis BIZOT, Damien CHANCEREL,  
Victor CHARDERON, Martin NIOMBELA, Adam  
SERRAKH

Enseignant(s) :

M. AUGER, M. MARTEL

Cours :

Analyse des besoins

# Sommaire

1	INTRODUCTION	3
2	LES FONCTIONNALITÉS DEMANDÉES	3
2.1	ACTEURS ET FINALITÉ	3
2.2	EXIGENCES	4
3	UTILISATIONS DE L'APPLICATION	6
3.1	DIAGRAMME DE CAS D'UTILISATION	6
3.2	SPÉCIFICATIONS DES CAS D'UTILISATION	6
4	SCÉNARIOS D'UTILISATION	10
5	EXIGENCES NON FONCTIONNELLES	12
6	PROTOTYPE DE L'INTERFACE UTILISATEUR	12
7	FICHES DE TESTS SYSTÈME	13
7.1	CRÉER UN SIMPLEXE	13
7.2	CHARGER UN SIMPLEXE	13
7.3	ENREGISTRER UN SIMPLEXE	14
7.4	ÉCHANGER DES VARIABLES	14
7.5	ANNULER UN ÉCHANGE DE VARIABLES	14
7.6	OBTENIR DES INDICATIONS SUR LE PROCHAIN ÉCHANGE	15
8	CONCLUSION	15
9	ANNEXE 1 : EXPLICATIONS SUR LE SIMPLEXE	17
10	ANNEXE 2 : EXEMPLE D'UN ÉCHANGE DÉTAILLÉ	18

# 1 Introduction

Dans le cadre de notre projet tutoré, nous avons été chargés par M. Martel (professeur de mathématiques) et M. Auger (professeur d'algorithmique et de mathématiques) de réaliser une application visant à permettre aux élèves du semestre 4 en DUT Informatique de dérouler la méthode mathématique du simplexe, en déchargeant la partie calculatoire (lourde et source d'erreur) sur l'ordinateur.

Ce document regroupe donc l'ensemble des besoins qui nous ont permis de concevoir une application répondant au mieux aux besoins de nos clients. Nous avons également détaillé les différentes utilisations possibles de notre logiciel, ainsi que les tests à réaliser pour déterminer si le produit fini correspond aux exigences.

## 2 Les fonctionnalités demandées

### 2.1 Acteurs et finalité

#### 2.1.1 Groupe

**Chef de projet** : Claire BAUCHU

**Membres** : Louis BIZOT, Damien CHANCEREL, Victor CHARDERON, Martin NIOMBELA, Adam SERRAKH

#### 2.1.2 Projet

Dans le cadre de notre projet tutoré, nous avons choisi le sujet « Interface Linéaire ». Ce projet est une application qui pourra être utilisée à des fins pédagogiques en mathématiques, dans les années à venir.

#### 2.1.3 Organisation et intervenants

IUT de Vélizy-Villacoublay (UVSQ)

**Clients** : M. Auger & M. Martel

#### 2.1.4 Caractéristiques générales et objectifs

- Type d'application : utilitaire

- Compatible sur PC
- Cibles : professeur et élèves

Le but du projet est de permettre aux élèves de comprendre l'algorithme du simplexe en s'extrayant des calculs nécessaires pour le résoudre, qui peuvent être source d'erreurs. Il est donc obligatoire de produire une application pédagogique, à savoir claire et intuitive.

L'application doit donc comporter une interface sur laquelle l'utilisateur peut résoudre un simplexe en effectuant les actions qu'il jugera adaptées. Il pourra ensuite conserver son travail, et recommencer autant de fois qu'il le souhaite. Puisque cette application est produite à des fins pédagogiques, il devra être possible de revenir en arrière après une action.

## 2.2 Exigences

Nous avons hiérarchisé la priorité de ces exigences selon cette échelle :

- **Critique** : importance capitale. Cette exigence constitue l'essence-même du projet
- **Majeur** : grande importance. Cette exigence est indispensable pour utiliser le projet final. Cependant, le système peut s'en passer au début.
- **Mineur** : optionnel. Le système fonctionne sans, mais cette exigence représente une option intéressante pour le projet.

### 2.2.1 Exigences critiques

#### 2.2.1.1 *Générer un simplexe*

L'utilisateur doit pouvoir créer un simplexe qu'il va pouvoir ensuite manipuler. La création d'un simplexe passe par le choix de plusieurs paramètres, détaillés dans les besoins suivants.

##### 2.2.1.1.1 Choisir le nombre de variables

L'utilisateur peut choisir de générer un simplexe avec différents nombres de variables et doit donc choisir ce nombre lors de la génération d'un simplexe. Il a été décidé avec M. Martel de limiter le nombre de variables à 7.

**Critère d'appréciation** : Le nombre de variables entré par l'utilisateur correspond au nombre de variables présentes dans le simplexe généré.

##### 2.2.1.1.2 Choisir les paramètres du simplexe

L'utilisateur doit pouvoir choisir quels paramètres il souhaite faire figurer dans son simplexe, à savoir les contraintes explicites et la fonction économique du simplexe.

**Critère d'appréciation** : La fonction économique correspond à celle entrée par l'utilisateur, et les contraintes explicites entrées sont les mêmes que celles affichées dans le simplexe ainsi généré.

##### 2.2.1.2 *Choisir d'échanger deux variables*

La manipulation d'un simplexe pour trouver des solutions admissibles passe par l'échange de variables. L'utilisateur doit donc pouvoir choisir deux variables à échanger.

**Critère d'appréciation** : L'utilisateur choisit deux variables à échanger en cliquant dessus, et le simplexe est mis à jour en prenant en compte l'échange ainsi effectué.

## 2.2.2 Exigences majeures

### 2.2.2.1 *Revenir en arrière / annuler une action*

L'échange de deux variables peut avoir été une erreur. L'utilisateur doit pouvoir cliquer sur un bouton qui lui permet de revenir en arrière en annulant l'action effectuée précédemment.

**Critère d'appréciation** : Un clic sur le bouton « revenir en arrière » affiche le simplexe tel qu'il était avant la dernière action effectuée. L'utilisateur peut cliquer autant de fois qu'il le souhaite sur le bouton, jusqu'à la première version du simplexe s'il le désire.

### 2.2.2.2 *Enregistrer un simplexe*

L'utilisateur doit pouvoir sauvegarder son travail. Ainsi, s'il décide d'enregistrer son travail, un fichier préalablement créé doit se mettre à jour et contenir le simplexe de l'utilisateur.

**Critère d'appréciation** : Le fichier est effectivement mis à jour et contient le simplexe de l'utilisateur. Il doit également contenir l'historique des actions effectuées sur le simplexe, pour que l'utilisateur qui réutilise postérieurement le fichier enregistré puisse annuler une action effectuée.

### 2.2.2.3 *Charger un simplexe enregistré*

L'utilisateur doit pouvoir charger un simplexe préalablement enregistré. Ainsi, il peut continuer ses manipulations ou revenir en arrière dans l'historique des actions. Pour cela, il sera proposé à l'utilisateur de parcourir ses fichiers pour qu'il sélectionne le fichier qu'il désire utiliser. Lorsque l'utilisateur valide son fichier, l'espace de travail se charge avec le simplexe que l'utilisateur a déjà manipulé par le passé ou non.

**Critère d'appréciation** : Le fichier du simplexe choisi correspond bien à celui qui est affiché dans l'espace de manipulation du simplexe. L'historique est toujours présent et l'utilisateur peut annuler une action effectuée avant la dernière sauvegarde du fichier.

## 2.2.3 Exigences mineures

### 2.2.3.1 *Obtenir des indices sur la marche à suivre durant la méthode du simplexe*

L'utilisateur doit pouvoir, en cliquant sur un bouton à tout moment de ses manipulations sur le simplexe, obtenir des indications sur la marche à suivre, sous la forme d'une indication de l'échange de variables prochain à effectuer pour obtenir le résultat optimal.

**Critère d'appréciation** : L'indication donnée s'affiche sur l'écran et correspond effectivement à l'échange le plus pertinent pour obtenir le meilleur résultat possible à partir du dernier simplexe obtenu par l'élève.

### 2.2.3.2 Zoomer sur une partie de l'écran

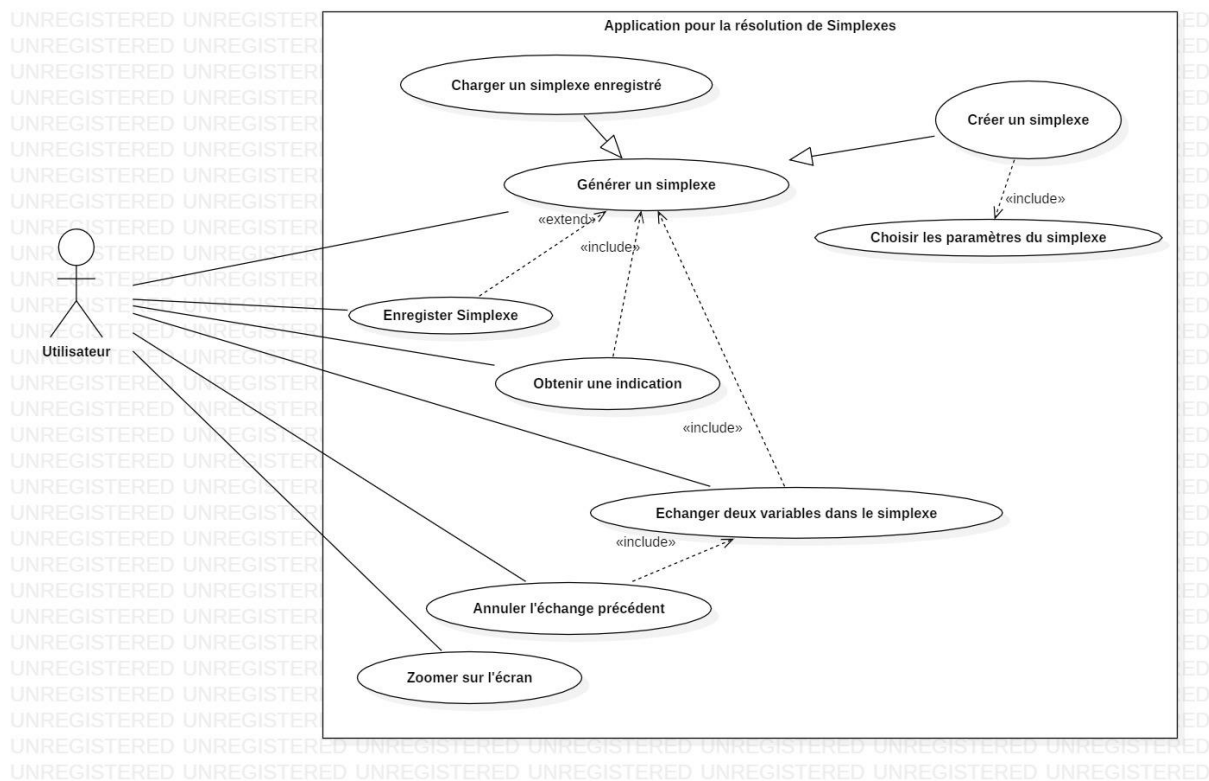
L'utilisateur doit pouvoir, au moyen d'un curseur, sélectionner une zone de l'écran à agrandir.

**Critère d'appréciation** : La zone zoomée est affichée dans une autre fenêtre et correspond effectivement à la zone visée par l'utilisateur.

## 3 Utilisations de l'application

### 3.1 Diagramme de cas d'utilisation

Le diagramme suivant est un diagramme de cas d'utilisation réalisé grâce au langage UML.



### 3.2 Spécifications des cas d'utilisation

Le niveau est déterminé selon cette échelle :

- **Utilisateur** : Fonction principale utilisée par l'utilisateur pour atteindre son objectif
- **Sous-fonctions** : Fonction ne pouvant se réaliser seule

### 3.2.1 Créer un simplexe

<i>Nom</i>	Créer un simplexe
<i>Contexte d'utilisation</i>	Pour créer un nouveau simplexe, et commencer à le manipuler ensuite
<i>Niveau</i>	Utilisateur
<i>Précondition</i>	Aucune
<i>Déclencheur</i>	Remplir le formulaire et cliquer sur « Créer »
<i>Scénario nominal</i>	L'utilisateur entre les différents paramètres qu'il souhaite fournir au simplexe, et clique sur « Créer ». Le simplexe est alors créé et peut être manipulé
<i>Variantes</i>	L'utilisateur a déjà créé un simplexe. Il peut en créer un nouveau en suivant les mêmes instructions que plus haut.

### 3.2.2 Charger un simplexe

<i>Nom</i>	Charger un simplexe
<i>Contexte d'utilisation</i>	Pour charger un ancien simplexe enregistré et pouvoir
<i>Niveau</i>	Utilisateur
<i>Précondition</i>	Posséder un simplexe enregistré au préalable dans ses fichiers sur son ordinateur
<i>Déclencheur</i>	Sélectionner un fichier à ouvrir après avoir atteint le menu « Charger un simplexe »
<i>Scénario nominal</i>	L'utilisateur parcourt ses dossiers jusqu'à trouver un fichier contenant un simplexe et pouvant être ouvert sur l'application. Il le sélectionne et clique sur « Charger ».
<i>Variantes</i>	L'utilisateur a déjà chargé ou créé un simplexe. Il peut en charger un nouveau en suivant les mêmes instructions que plus haut.

### 3.2.3 Enregistrer un simplexe

<i>Nom</i>	Enregistrer un simplexe
<i>Contexte d'utilisation</i>	Un simplexe a été créé ou chargé. L'utilisateur l'a manipulé ou non. Il veut

	cependant garder le simplexe tel qu'il est à l'instant T (ainsi que l'historique de ces actions) pour pouvoir charger ce simplexe plus tard et le retrouver tel qu'à l'instant T.
<i>Niveau</i>	Sous-fonction
<i>Précondition</i>	Avoir chargé ou créé un simplexe (si l'utilisateur n'a pas sélectionné de simplexe sur lequel travailler, rien ne s'enregistre)
<i>Déclencheur</i>	Cliquer dans la barre en haut de l'écran sur « Enregistrer »
<i>Scénario nominal</i>	L'utilisateur a ouvert un simplexe (créé ou chargé) et l'enregistre en cliquant sur « Enregistrer ». Il peut trouver le simplexe dans ses dossiers ensuite.
<i>Variantes</i>	-

### 3.2.4 Échanger deux variables

<i>Nom</i>	Echanger deux variables
<i>Contexte d'utilisation</i>	Un simplexe est chargé sur l'application (créé ou chargé depuis les dossiers). L'utilisateur souhaite échanger deux variables dans le simplexe (voir Annexe 1).
<i>Niveau</i>	Utilisateur
<i>Précondition</i>	Avoir chargé un simplexe sur lequel travailler
<i>Déclencheur</i>	Un clic sur une inconnue dans le dans le dernier simplexe calculé.
<i>Scénario nominal</i>	L'utilisateur clique sur une inconnue pour la sélectionner. L'inconnue égale à la ligne contenant l'inconnue cliquée va être échangée avec cette dernière. Un nouveau simplexe se charge avec les deux variables échangées.
<i>Variantes</i>	-

### 3.2.5 Annuler l'échange précédent

<i>Nom</i>	Annuler l'échange précédent
------------	-----------------------------



*Contexte d'utilisation*

L'utilisateur souhaite revenir à l'ancien simplexe, avant l'échange de variables qu'il vient d'effectuer

*Niveau*

Sous-fonction

*Précondition*

Avoir chargé un simplexe sur lequel travailler, et avoir effectué un échange de variables.

*Déclencheur*

Clic sur « revenir en arrière » dans la barre de menu en haut de la page.

*Scénario nominal*

L'utilisateur clique sur « revenir en arrière ». Le dernier simplexe calculé est supprimé, et l'utilisateur peut effectuer l'échange qu'il souhaite sur le simplexe tel qu'il était avant l'échange.

*Variantes*

L'utilisateur peut appuyer plusieurs fois sur « revenir en arrière », jusqu'à retourner au premier simplexe s'il le souhaite.

## 3.2.6 Zoomer sur l'écran

*Nom*

Zoomer sur l'écran

*Contexte d'utilisation*

L'utilisateur souhaite agrandir une zone de l'écran

*Niveau*

Utilisateur

*Précondition*

-

*Déclencheur*

Cliquer sur la loupe dans la barre des tâches, puis cliquer sur une zone de l'écran

*Scénario nominal*

L'utilisateur clique sur la loupe puis sur une zone de l'écran. Une fenêtre s'ouvre alors contenant un agrandissement de la zone autour de l'endroit où l'utilisateur a cliqué.

*Variantes*

-

## 3.2.7 Obtenir une indication sur le prochain échange

*Nom*

Obtenir une indication sur le prochain échange

*Contexte d'utilisation*

L'utilisateur souhaite obtenir un indice sur le prochain échange pour obtenir un résultat final optimal (voir Annexe 1).

*Niveau*

Sous-fonction

*Précondition*

*Déclencheur*

*Scénario nominal*

*Variantes*

Un simplexe est chargé sur l'application  
(créé ou chargé depuis les dossiers)

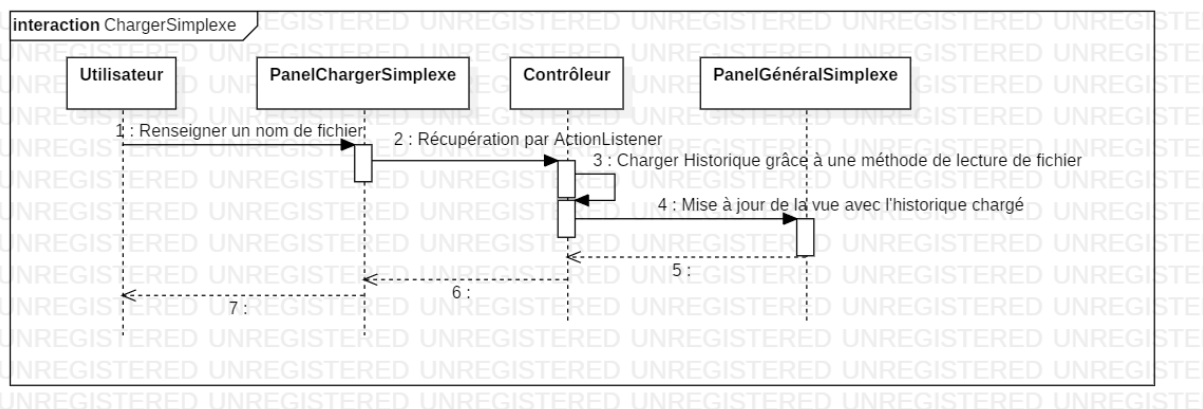
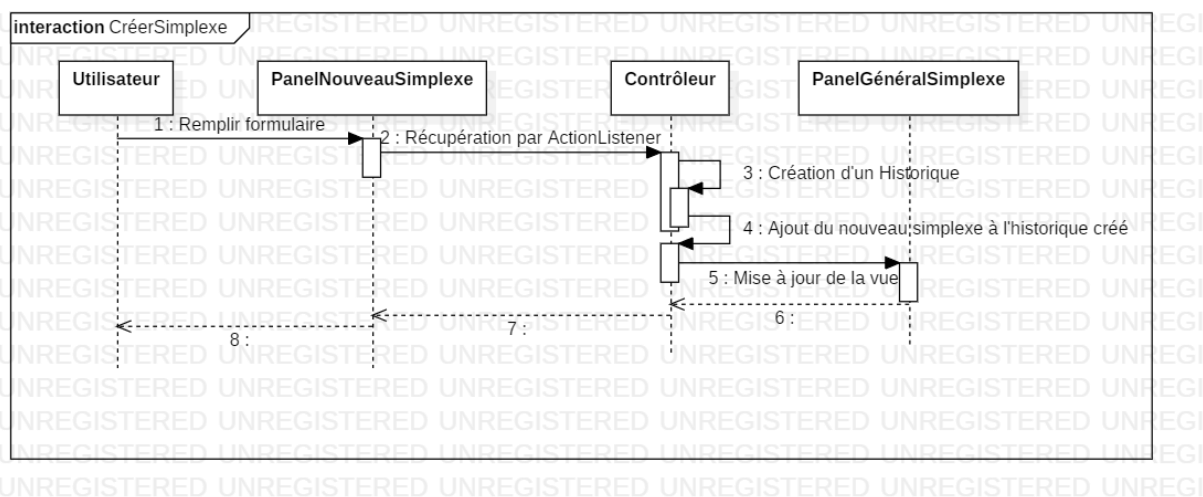
Clic sur un bouton à côté du dernier  
simplexe calculé

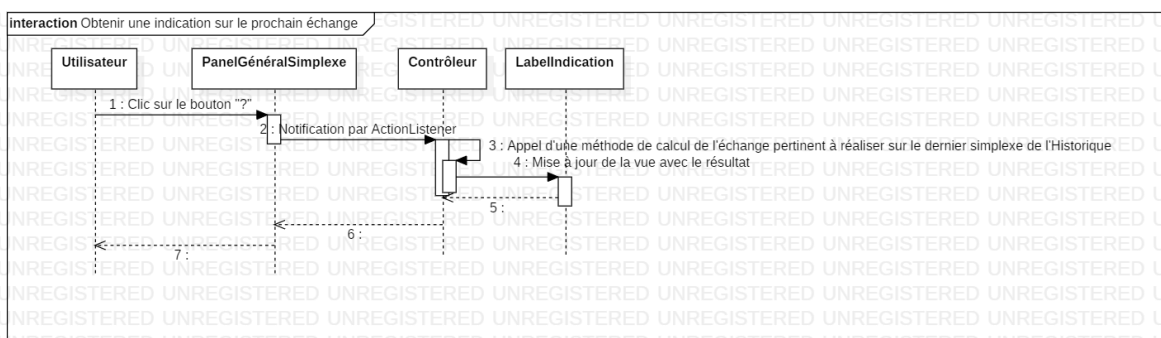
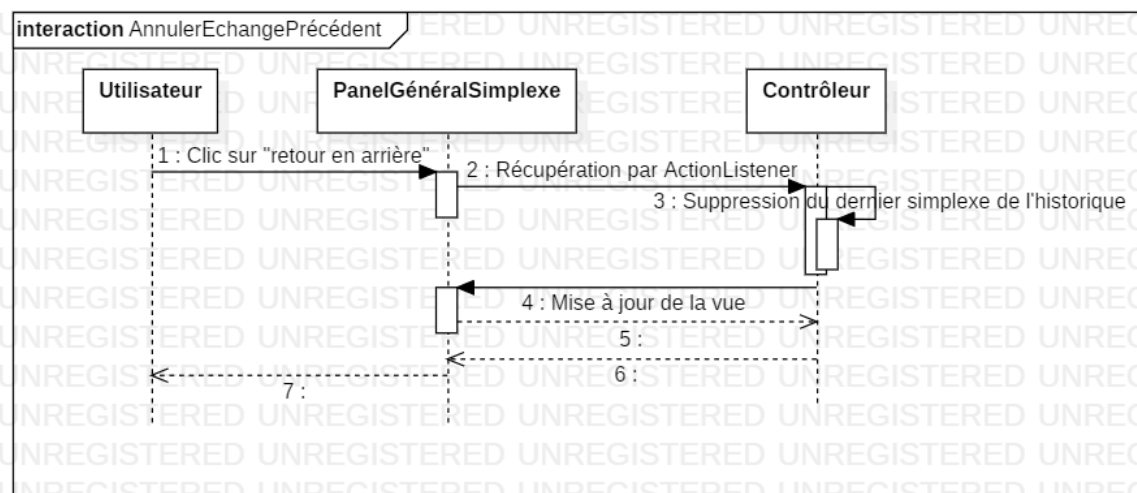
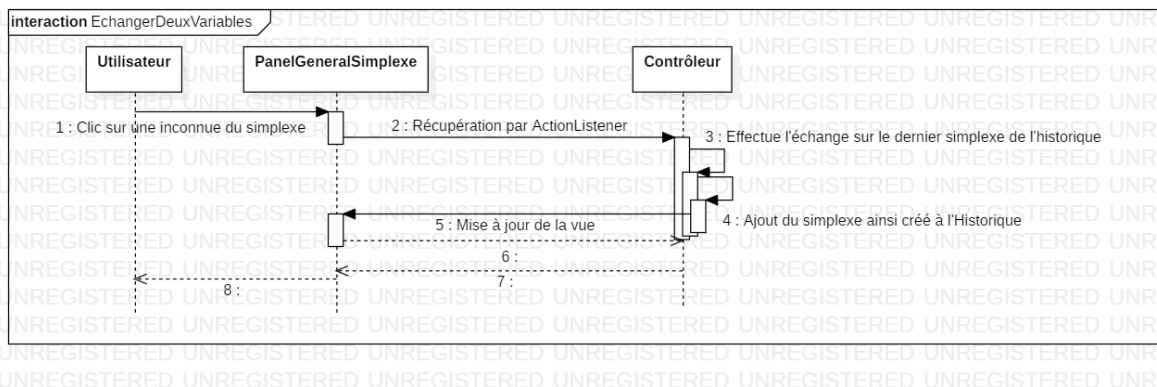
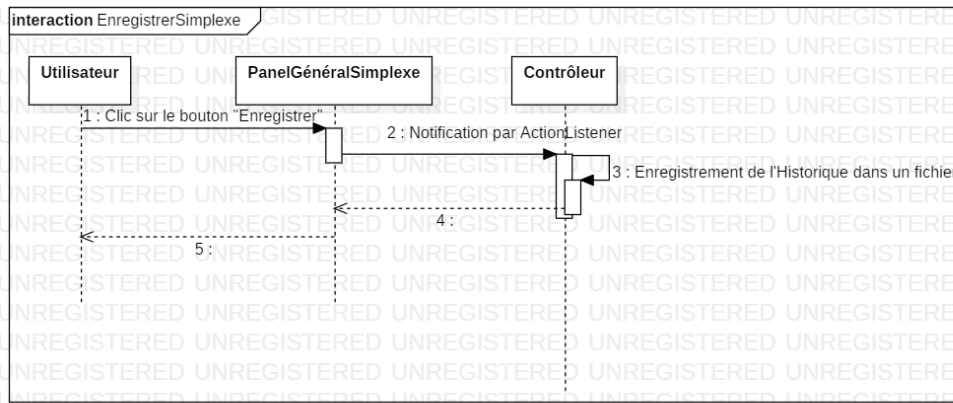
L'utilisateur clique sur le bouton prévu à  
cet effet, et une indication s'affiche sur  
une zone de l'écran. Cette indication se  
présente sous la forme d'un échange à  
effectuer ensuite. Cet échange est  
calculé pour obtenir le meilleur résultat  
possible à partir du dernier simplexe  
calculé.

-

## 4 Scénarios d'utilisation

Les diagrammes suivants sont des diagrammes de séquence illustrant les différents cas d'utilisation.



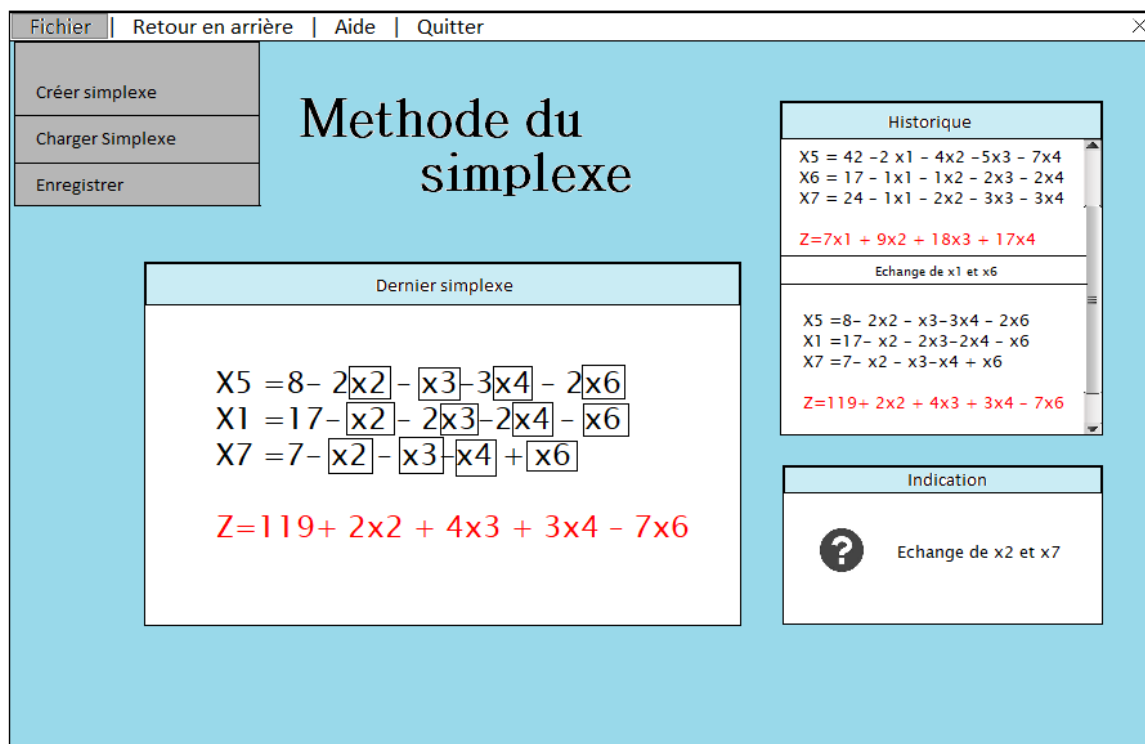


# 5 Exigences non fonctionnelles

L'application doit être utilisable sur ordinateur. Nous nous faisons également un devoir de la rendre la plus intuitive possible, puisqu'elle est destinée à des élèves et des professeurs. Par ailleurs, nous souhaitons la rendre pratique à utiliser, notamment par l'utilisation de mnémoniques (raccourcis clavier) et d'autres fonctionnalités, comme un historique des échanges de variables effectués, et un lien vers le simplexe concerné si l'on clique sur un échange effectué auparavant dans l'historique.

Ces fonctionnalités ne sont pas celles que nous implémenterons en premier, mais figurent parmi les exigences non-fonctionnelles.

## 6 Prototype de l'interface utilisateur



Le panel dernier simplexe comporte le dernier simplexe obtenu au fur et à mesure des échanges, et les carrés autour des inconnues représentent des boutons cliquables. Si l'on clique sur le bouton x2 de la première ligne du simplexe, l'échange effectué est celui-ci : x2 et x5.

En cliquant sur Créer simplexe ou charger simplexe, on accède à deux formulaires différents décrits dans le dossier concepteur.

## 7 Fiches de tests système

### 7.1 Créer un simplexe

<b>Identification du test :</b> Créer Simplexe	
<b>Description du test :</b> L'utilisateur entre le nombre de contraintes et de monômes qu'il souhaite voir apparaître dans le simplexe qu'il va utiliser. Il remplit ensuite les champs pour ajouter des coefficients aux inconnues du simplexe ainsi que dans la fonction économique, ainsi que la contrainte de chaque ligne (après le $\leq$ ). En cliquant sur « Créer », un simplexe correspondant aux valeurs entrées doit être créé.	
<b>Ressources requises :</b> aucune	
<b>Entrées :</b> Nombre de contraintes : 2 Nombre de monômes : 2 Coefficients entrés : $30x_1 + 20x_2 \leq 180$ $30x_1 + 40x_2 \leq 140$ $Z = 50x_1 + 20x_2$	
<b>Résultat attendu :</b> Simplexe à manipuler = $30x_1 + 20x_2 \leq 180$ $30x_1 + 40x_2 \leq 140$ $Z = 50x_1 + 20x_2$	

### 7.2 Charger un simplexe

<b>Identification du test :</b> Charger Simplexe	
<b>Description du test :</b> L'utilisateur choisit un fichier dans son ordinateur contenant un simplexe préalablement créé et enregistré. A l'ouverture, le simplexe contenu dans le fichier est lu puis implémenté dans l'application	
<b>Ressources requises :</b> un fichier contenant un simplexe enregistré	
<b>Entrées :</b> nom du fichier désiré	
<b>Résultat attendu :</b> Simplexe à manipuler = simplexe contenu dans le fichier	

## 7.3 Enregistrer un simplexe

<b>Identification du test :</b> Enregistrer Simplexe	
<b>Description du test :</b> L'utilisateur enregistre le simplexe qu'il manipule	
<b>Ressources requises :</b> aucune	
<b>Entrées :</b> un simplexe	
<b>Résultat attendu :</b> création d'un nouveau fichier si c'est la première fois que le simplexe est enregistré, sinon suppression de l'ancien fichier et en recreation du fichier avec le simplexe modifié. Le fichier peut être ré-ouvert dans l'application (voir Test Charger Simplexe)	

## 7.4 Echanger des variables

<b>Identification du test :</b> Echanger variables	
<b>Description du test :</b> L'utilisateur échange deux variables (ici, $x_3$ et $x_7$ )	
<b>Ressources requises :</b> un simplexe	
<b>Entrées :</b> $\begin{cases} x_5 = 42 - 2x_1 - 4x_2 - 5x_3 - 7x_4 \\ x_6 = 17 - x_1 - x_2 - 2x_3 - 2x_4 \\ x_7 = 24 - x_1 - 2x_2 - 3x_3 - 3x_4 \\ z = 7x_1 + 9x_2 + 18x_3 + 17x_4 \end{cases}$	
<b>Résultat attendu :</b> $\begin{cases} x_3 = 8 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - x_4 - \frac{1}{3}x_7 \\ x_5 = 2 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - 2x_4 + \frac{5}{3}x_7 \\ x_6 = 1 - \frac{1}{3}x_1 + \frac{1}{3}x_2 + \frac{2}{3}x_7 \\ z = 144 + x_1 - 3x_2 - x_4 - 6x_7 \end{cases}$	

## 7.5 Annuler un échange de variables

<b>Identification du test :</b> Annuler échange	
<b>Description du test :</b> L'utilisateur a effectué un échange, et il souhaite annuler celui-ci pour revenir au résultat précédent	
<b>Ressources requises :</b> un simplexe sur lequel un échange a été effectué	
<b>Entrées :</b> On vient de faire l'échange $x_3$ et $x_7$ . On a obtenu : $\begin{cases} x_3 = 8 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - x_4 - \frac{1}{3}x_7 \\ x_5 = 2 - \frac{1}{3}x_1 - \frac{2}{3}x_2 - 2x_4 + \frac{5}{3}x_7 \\ x_6 = 1 - \frac{1}{3}x_1 + \frac{1}{3}x_2 + \frac{2}{3}x_7 \\ z = 144 + x_1 - 3x_2 - x_4 - 6x_7 \end{cases}$	

Résultat attendu :

$$\begin{cases} x_5 = 42 - 2x_1 - 4x_2 - 5x_3 - 7x_4 \\ x_6 = 17 - x_1 - x_2 - 2x_3 - 2x_4 \\ x_7 = 24 - x_1 - 2x_2 - 3x_3 - 3x_4 \\ z = 7x_1 + 9x_2 + 18x_3 + 17x_4 \end{cases}$$

## 7.6 Obtenir des indications sur le prochain échange

**Identification du test :** Obtenir des indications sur le prochain échange

**Description du test :** L'utilisateur ne sait pas quel échange faire ensuite pour obtenir un résultat optimal dans la fonction économique

**Ressources requises :** un simplexe

**Entrées :**

$$\begin{cases} x_5 = 42 - 2x_1 - 4x_2 - 5x_3 - 7x_4 \\ x_6 = 17 - x_1 - x_2 - 2x_3 - 2x_4 \\ x_7 = 24 - x_1 - 2x_2 - 3x_3 - 3x_4 \\ z = 7x_1 + 9x_2 + 18x_3 + 17x_4 \end{cases}$$

**Résultat attendu :** « Il faut faire l'échange  $x_3$  et  $x_7$  »

# 8 Conclusion

Ce document nous a permis de détailler les attentes de nos clients M.Auger et M.Martel. Nous avons donc pu prioriser les besoins et ainsi établir un programme pour notre développement, ainsi que la répartition des tâches au sein de notre groupe.

Nous avons également déterminé les différents tests à réaliser pour valider les fonctionnalités. Ce dossier nous a donc permis d'établir le squelette de notre logiciel, grâce aux scénarios d'utilisation.

Effectuer ces recherches et études préalables nous permettront d'établir une conception plus juste, et d'ainsi éviter des erreurs dès le début de notre développement, pouvant mener à l'échec du projet.

# Annexe : Simplexe

---



# 9 Annexe 1 : Explications sur le simplexe

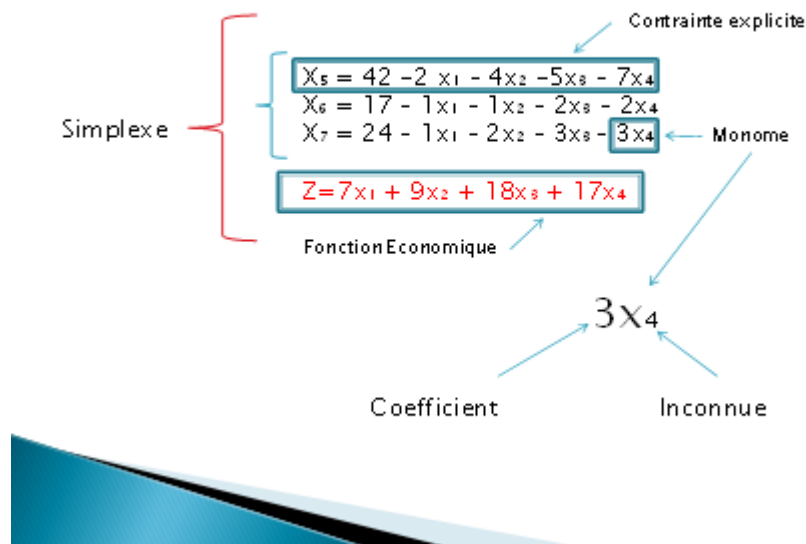


Image 1 : Schéma explicatif d'un simplexe

Un simplexe est une structure comprenant des inconnues. Cette structure permet notamment de modéliser des situations où l'on souhaite calculer un bénéfice. Le but de la méthode du simplexe est d'obtenir le plus grand nombre (sans inconnue) dans la fonction économique. Cependant, il faut choisir le bon échange à effectuer. De mauvais échanges (ou de bons échanges dans le mauvais ordre) mènent à un résultat possible, mais non optimal. (Exemple : le résultat optimal de ce simplexe est indiqué en Annexe 2. C'est 147, premier nombre de la fonction économique).

# 10 Annexe 2 : Exemple d'un échange détaillé

$$\begin{cases} X_5 = 42 - 2x_1 - 4x_2 - 5x_3 - 7x_4 \\ X_6 = 17 - 1x_1 - 1x_2 - 2x_3 - 2x_4 \\ X_7 = 24 - 1x_1 - 2x_2 - 3x_3 - 3x_4 \end{cases} \quad \text{Simplexe de départ}$$

$$Z = 7x_1 + 9x_2 + 18x_3 + 17x_4$$

$$\begin{aligned} X_5 &= 42 - 2(17 - x_2 - 2x_3 - 2x_4 - x_6) - 4x_2 - 5x_3 - 7x_4 \\ X_6 &= 17 - x_2 - 2x_3 - 2x_4 - x_6 \\ X_7 &= 24 - (17 - x_2 - 2x_3 - 2x_4 - x_6) - 2x_2 - 3x_3 - 3x_4 \end{aligned}$$

$$\begin{cases} X_5 = 8 - 2x_2 - x_3 - 3x_4 - 2x_6 \\ X_1 = 17 - x_2 - 2x_3 - 2x_4 - x_6 \\ X_7 = 7 - x_2 - x_3 - x_4 + x_6 \end{cases} \quad \text{Echange de } x_1 \text{ et } x_6$$

$$Z = 119 + 2x_2 + 4x_3 + 3x_4 - 7x_6 \quad \text{Résultat à obtenir}$$

$$\begin{aligned} X_1 &= 3 + x_2 - 3x_3 + 2x_7 \\ X_5 &= 1 - 1x_2 - x_4 + x_6 - x_7 \\ X_6 &= 7 - x_2 - 2x_4 + x_6 + x_7 \end{aligned}$$

$$Z = 147 - 2x_2 - x_4 - 3x_6 - 4x_7$$

Image 2 : Exemple d'un échange dans un simplexe

La partie en vert représente l'ensemble des calculs à effectuer à l'écrit pour obtenir le simplexe après l'échange de  $x_1$  et  $x_6$ . Notre application doit donc éviter ces calculs. Dans cette dernière, on obtiendra immédiatement le résultat du simplexe après l'échange. Il faudra donc effectuer plusieurs échanges, dans le bon ordre (et sans erreur de calculs si on résout par écrit), pour arriver au résultat à obtenir, qui est le résultat optimal.