

Juin 2020

Rapport Festi'Go

L'histoire d'une application de gestion de festival



**Glory Sara Tairi - Justine Fouillé
Paul Breton - Corentin Plestan - Melvin Mathurin**

Sommaire

Introduction	2
Gestion de projet	3
Problématique	3
Figure 1 : Bête à corne	3
Fiche de définition	3
Contexte	3
Principaux éléments de l'analyse fonctionnelle	4
Principaux risques	5
Acteurs	5
Diagramme des tâches (WBS)	5
Diagramme de responsabilités (OBS)	6
Matrice RACI	6
Vraie gestion de projet	7
Diagramme de GANTT	7
Diagramme des tâches (WBS)	7
Diagramme de responsabilités (OBS)	8
Conception	8
Maquette	8
Schéma navigationnel d'interaction (SNI)	10
Diagramme de classes	12
Explication des algorithmes	13
Algorithme 1 : Placement des éléments dans l'espace	13
But : Placer un composant (Stand, Scène) dans l'espace.	13
Algorithme 2 : Sérialisation des éléments	13
Réalisation	14
La documentation utilisateur	15
Conclusion	16
Table des d'illustrations	17

1. Introduction

“Lorsque deux forces sont jointes, leur efficacité est double”

Isaac Newton

5 étudiants, 3 mois, 1 projet. Dans ce rapport vous trouverez tous les détails du projet de notre application Festi’Go. Son objectif est d’aider à l’organisation d’un festival, notamment par rapport à la gestion géographique du site.

Dans le cahier des charges, il est demandé une application gérant la position du festival dans une ville, mais également la disposition des éléments de celui-ci sur le site géographique du festival. De manière plus précise, la gestion des emplacements doit se faire par le biais d’une grille sur laquelle l’utilisateur positionne les différents stands, scènes, et autres. Ceux-ci devront avoir une taille précise.

Ensuite, il doit également être possible de saisir les artistes qui feront une représentation avec leur genre musical et la durée de leur prestation. Cela permet d’afficher un planning.

L’application doit également être en mesure de calculer les recettes du festival avec l’entrée des tarifs des billets et des diverses dépenses qu’engendre le festival (optionnel).

Après une étude du cahier des charges, nous avons dû faire des choix pour essayer de le respecter au mieux. Nous avons donc décidé par exemple d’initialiser directement dans l’application des villes. L’utilisateur peut donc choisir entre plusieurs dispositions différentes celle qui lui convient le mieux. Nous avons également pris la décision de faire une application comportant plusieurs écrans peu importe combien de travail cela demande. Ainsi, il faut donc penser à un menu principal pour assurer la navigation entre tous les éléments demandés dans le cahier des charges. De plus, nous avons également décidé de gérer les dépenses et les recettes du festival. Enfin, nous avons appliqué des contraintes concernant le festival dans son ensemble. La durée du festival ne peut pas dépasser les trois jours par exemple.

Dans un premier temps vous verrez comment nous avons organisé notre projet. Nous aborderons notamment la fiche de définition (WBS, OBS, diagramme de Gantt...) mais nous vous expliquerons également comment cela s’est vraiment passé et quelles ont été nos difficultés par rapport à ce qui était prévu.

Ensuite, nous vous parlerons de la partie conception du projet. Nous présentons la maquette que nous avons imaginé, le SNI qui expliquera les aspects dynamiques de celle-ci, puis le diagramme de classe.

Enfin, par le biais d’une présentation des fonctionnalités et des documentations utilisateurs et technique (JavaDoc) nous vous parlerons de la conception de notre logiciel.

2. Gestion de projet

Dans cette partie nous allons vous détailler les différents points de notre projet. Nous commencerons par les enjeux de celui-ci. Puis viendra l'analyse fonctionnelle. Ensuite, nous vous détaillerons les principaux risques que nous avons identifiés. Enfin, nous présenterons les acteurs de cette application. Pour finir, vous aurez une vue plus globale du projet comprenant les différents diagrammes : OBS, WBS, matrice RACI...

2.1. Problématique

Tout d'abord, nous avons dû définir l'objectif principal du projet. Pour illustrer cela nous avons réalisé une bête à corne. Nous avons commencé par identifier les bénéficiaires de l'application, dans notre cas les organisateurs d'un festival de musique. Puis ce sur quoi elle agirait, l'organisation du festival. Enfin son but principal, ici, est d'adapter l'organisation géographique (scènes, stands...) à un environnement donné.

Comment imaginer et développer une application qui mènera à bien l'organisation d'un festival ?

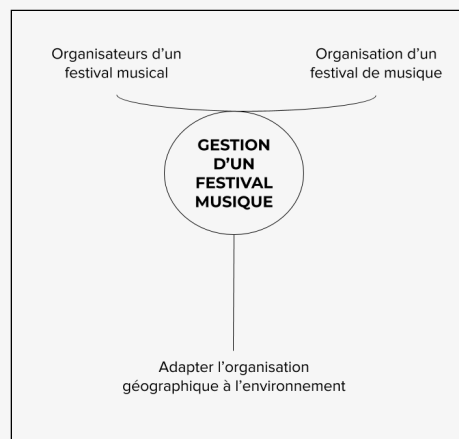


Figure 1 : Bête à corne

2.2. Fiche de définition

2.2.1. Contexte

Passons au contexte de l'application. Nous avons défini l'équipe comme client. Nous sommes donc des développeurs qui organisent un festival musical sur deux, trois jours, nommé "Java Landers", et avons décidé de concevoir et programmer une application pour nous aider dans notre quête. Par souci d'optimisation, nous avons pris la décision de coder une application globale qui pourra être réutilisable à souhait si nous organisons d'autres festivals. Nos partenaires sont diverses : le Télégramme, le Crédit Mutuel Bretagne, Henaff, et pour finir la Région Bretagne. Ils soutiennent tout d'abord le festival mais également le développement de l'application qu'ils pourront utiliser lors de leurs événements respectifs. Pour le lieu du festival, nous avons trouvé et loué un champ aux alentours de Lorient. Il est de forme rectangulaire, ce qui nous facilite la tâche pour concevoir la partie où l'on place les éléments sur la carte.

2.2.2. Principaux éléments de l'analyse fonctionnelle

Ensuite, nous avons identifié les différents besoins, qu'ils soient fonctionnels ou non fonctionnels.

- besoins fonctionnels
 - Affichage de la programmation du festival
 - ↳ récupération et triage des informations par le système
 - Personnalisation des caractéristiques du festival
 - ↳ emplacement du site, artistes, nombre de places, de scènes et de stands
 - Disposition des éléments du festival
 - ↳ transformé sous forme de carré en fonction des dimensions par le système
 - Calcul des recettes
 - ↳ tarif enfant, jeune, normal
 - ↳ coût du festival (artistes, scènes, techniciens, sécurité...)

- besoins non fonctionnels
 - Performance
 - ↳ Logiciel rapide
 - Convivialité
 - ↳ Ergonomie → Simplicité d'utilisation et d'affichage
 - ↳ Documentation
 - Maintenabilité
 - ↳ SAV (support téléphonique et technique)

Définissons les contraintes. Au fur et à mesure du mois de mars, une nouvelle s'est imposée à nous. Nous avons d'abord les contraintes qui concernent l'application en elle-même :

- Gestion des informations via un système de fichiers
- Application monoposte

Ensuite, celle qui concerne l'utilisateur :

- Interaction avec l'utilisateur (utilisation de bouton, saisie de texte...)

Et enfin, une qui s'applique plutôt à nos conditions de travail : le confinement dû à la crise sanitaire du coronavirus.

Pour finir, après avoir déduit tous ces aspects, il ne nous restait plus qu'à trouver les exclusions de notre projet. Les éléments techniques que nous n'allions pas utiliser. Ils concernent notamment la sauvegarde des informations qui ne se fera pas par une base de donnée mais par un système de fichier. Mais aussi l'accès à l'application qui ne se fera pas via internet puisque celle-ci est monoposte.

2.2.3. Principaux risques

Nous avons identifié les risques rapidement. Leur anticipation nous permet donc de réfléchir à des solutions plausibles pour y remédier. Les risques qui s'appliquent à la gestion du projet sont les suivants :

- La diminution ou manque d'effectifs qui aurait des conséquences sur la productivité générale

- La mauvaise gestion du projet (temps, outils, effectifs) qui nous mettrait en retard sur le projet
- La crise sanitaire qui a commencé en mars 2020. Celle-ci nous amène à travailler à distance avec nos coéquipiers
- La mauvaise communication entre les membres de l'équipe

2.2.4. Acteurs

Dans un avenir proche, les acteurs sont composés de l'équipe de gestion de projet puisqu'elle organise le festival et développe l'application pour ses propres besoins. Il n'y a donc pas de notions immédiates de client ou de superviseur.

👉 Equipe :

1. Paul Breton → Chef de projet
2. Corentin Plestan
3. Glory-Sara Tairi
4. Justine Fouillé
5. Melvin Mathurin

2.2.5. Diagramme des tâches (WBS)

Nous avons réalisé le diagramme de tâches sur draw.io. Après avoir tenté de découper chaque étape en des lots réalisables nous avons commencé à lier le tout.

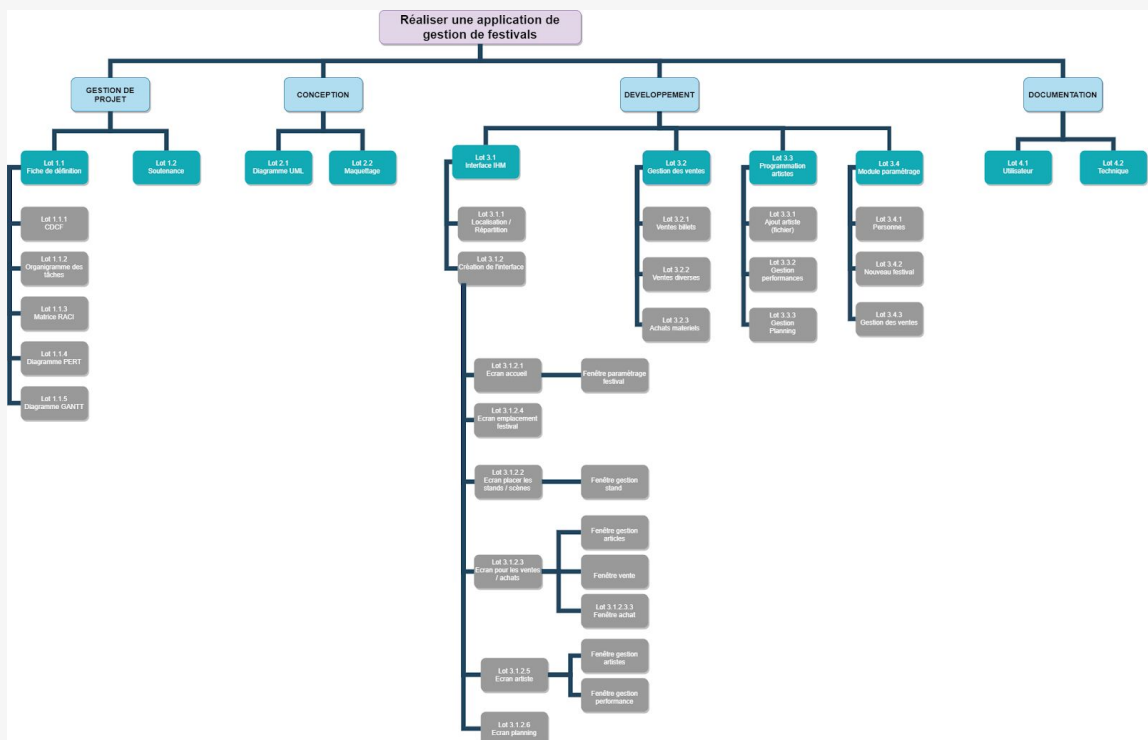


Figure 2 : Diagramme de tâches

2.2.6. Diagramme de responsabilités (OBS)

Nous avons également réalisé le diagramme de responsabilité sur draw.io. Celui-ci découle directement du diagramme des tâches.

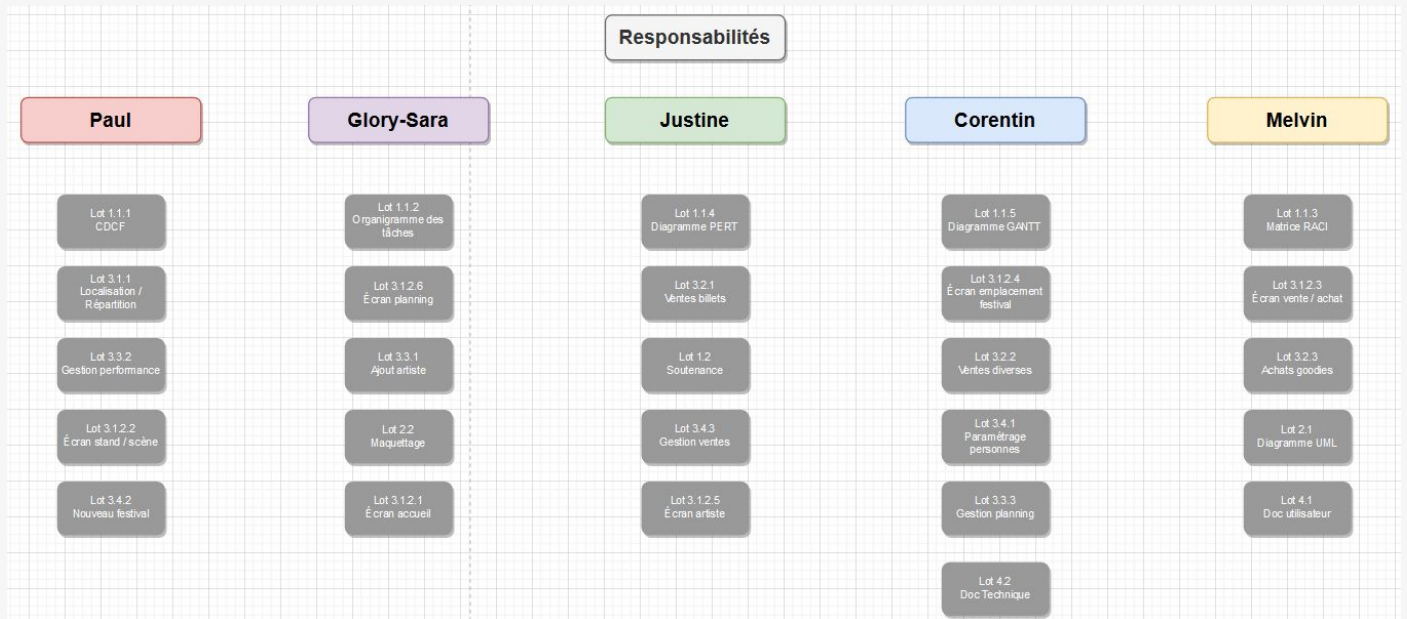


Figure 3 : Diagramme de responsabilités

2.2.7. Matrice RACI

Enfin, nous avons élaboré la matrice RACI sur Google Draw à partir du diagramme de responsabilité préalablement défini. Nous avons fait le choix de ne pas intégrer directement la première rangée de lots et de passer à la première rangée de sous lots.

Lot	Sous-lot	Paul	Corentin	Glory	Melvin	Justine
Lot 1	Lot 1.1.1	RA	A	A	A	A
	Lot 1.1.2	C	A	RA	C	C
	Lot 1.1.3	A	A	C	RA	A
	Lot 1.1.4	C	C	C	A	RA
	Lot 1.1.5	C	RA	C	A	I
Lot 2	Lot 1.2	A	A	A	A	RA
	Lot 2.1	C	I	C	RA	I
Lot 3	Lot 2.2	C	C	RA	C	A
	Lot 3.1.1	R	I	I	A	C
	Lot 3.1.2.1	I	I	RA	I	C
	Lot 3.1.2.2	RA	A	I	C	I
	Lot 3.1.2.3	I	C	I	RA	A
Lot 4	Lot 3.1.2.4	I	RA	C	I	C
	Lot 3.1.2.5	C	I	C	A	RA
	Lot 3.1.2.6	C	A	RA	I	I
	Lot 3.2.1	A	I	C	I	R
	Lot 3.2.2	I	R	I	A	C
Lot 4	Lot 3.2.3	C	A	C	R	I
	Lot 3.3.1	C	C	R	C	A
	Lot 3.3.2	R	C	I	I	C
	Lot 3.3.3	I	RA	A	C	C
	Lot 3.4.1	C	R	C	I	C
Lot 4	Lot 3.4.2	RA	I	C	C	I
	Lot 3.4.3	C	C	A	C	R
	Lot 4.1	I	A	A	R	A
Lot 4	Lot 4.2	C	R	I	A	A

2.3. Vraie gestion de projet

2.3.1. Diagramme de GANTT

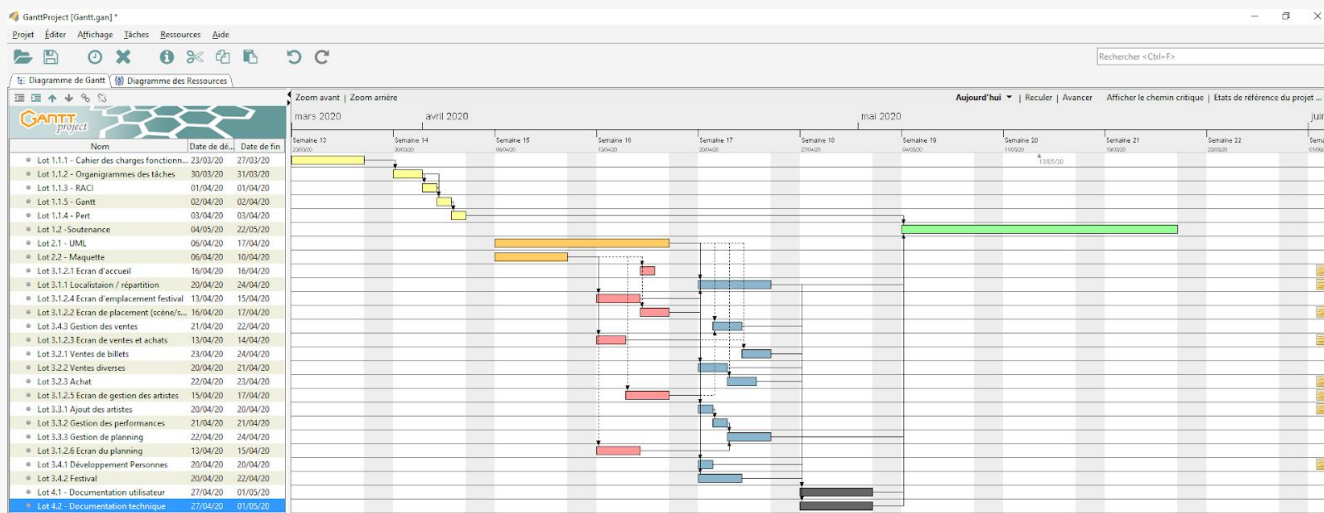


Figure 5 : Diagramme de Gantt

Notre incompréhension de JavaFX a eu un impact sur le planning, et donc le diagramme de Gantt. En effet, la plupart des écrans codés ont nécessités la création directe des classes associées. Nous pensions finir la programmation le plus tôt possible pour avoir le temps de faire la soutenance et le rapport dès le début du mois de juin, or, l'application n'est pas encore terminée et nous avançons déjà dans l'écriture du rapport.

Nous avons commencé à coder pendant les vacances d'avril. C'était une étape compliquée pour nous car nous n'avions pas encore eu de cours ni de travaux pratiques d'IHM. Nous n'avions aucune idée de comment nous allions coder ceci. Nous avons cherché des tutoriels, et des cours sur internet pour avoir une idée de ce vers quoi nous devons nous diriger.

La confection de la documentation a pris énormément de retard. Il est compliqué d'écrire une documentation lorsque le logiciel n'est pas encore finalisé.

2.3.2. Diagramme des tâches (WBS)

Lors de la gestion de projet, nous avons imaginé l'application sans aucune expérience de JavaFX. Nous avons donc développé des lots qui se sont révélés lors de la programmation inutiles. Dans un premier temps nous avons pris le soin de séparer la programmation IHM et la programmation orientée objet, or, nous nous sommes vite rendu compte que pour la plupart des aspects de l'application, ces deux parties allaient de pair. Par exemple, pour la partie des ventes, la gestion des ventes a été programmée en même temps que l'écran qui y est associé. La gestion des artistes a suivi le même processus. Nous avons donc surestimé le découpage des lots et compliqué le WBS et l'OBS. À l'avenir, nous nous renseignerons davantage sur les technologies que nous serons amenés à utiliser lors d'un projet.

2.3.3. Diagramme de responsabilités (OBS)

Comme expliqué précédemment, le mauvais découpage des lots a impacté la répartition du travail. De plus, certains lots ont pris plus de temps et ont été plus difficile à gérer que ce à quoi nous nous attendions. De par ce fait, la répartition des tâches a évolué au cours de la conception. Par exemple, la gestion des artistes a été développée par une seule personne au lieu de 3, il est en de même pour la gestion des dépenses/ventes.

Nous avons vu que la gestion d'un projet n'est pas une tâche facile. Entre la répartition des rôles, les risques rencontrés et une multitude d'autres facteurs, notre esprit d'équipe fut important pour renforcer les aspects négatifs et pour nous améliorer.

Nous allons passer à la conception de notre logiciel c'est-à-dire à toutes les étapes d'imagination pour notre future application.

3. Conception

La conception est une des étapes clé dans la création d'un logiciel. Elle permet de nous projeter et d'avoir une idée précises des étapes qu'il faudra mettre en place dans le développement de celui-ci.

Ici nous parlerons dans un premier temps la maquette que nous avons réalisé, accompagnée du SNI pour expliquer les aspects dynamiques de celle-ci.

Ensuite, nous présenterons le diagrammes de classes avec toutes les données utilisées dans le logiciel.

Puis pour finir, nous aborderons les différents algorithmes développés dans l'application, qu'ils soient simples ou plus complexes.

3.1. Maquette

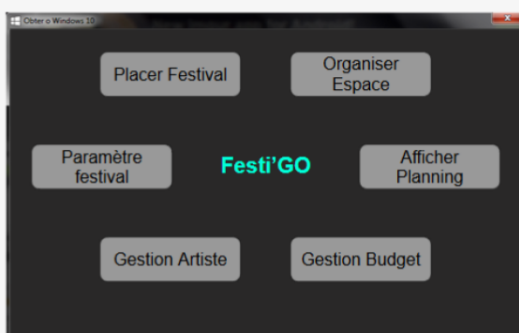


Figure 6 : Écran d'accueil de l'application Festi'GO

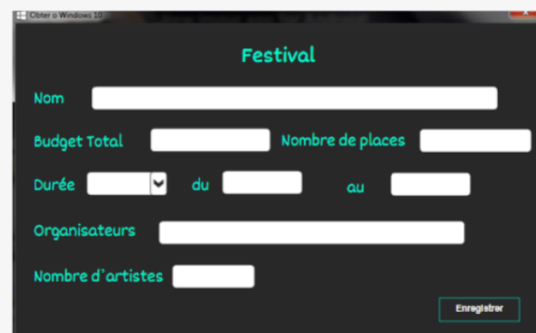


Figure 7 : Écran de création d'un festival

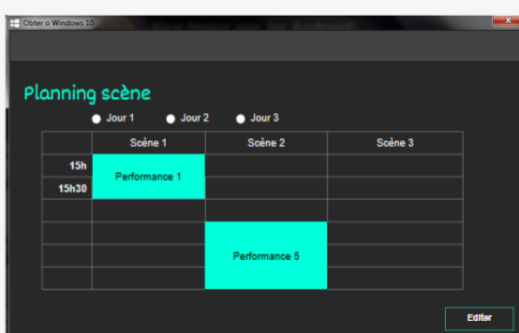


Figure 8 : Écran d'affichage des plannings

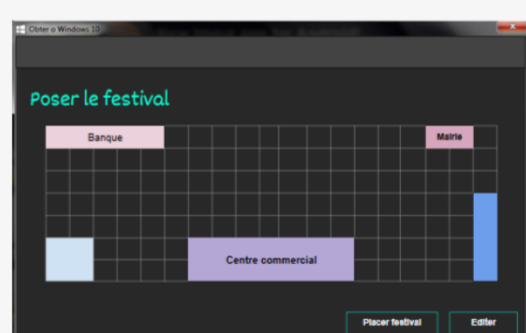


Figure 9 : Écran pour poser le festival

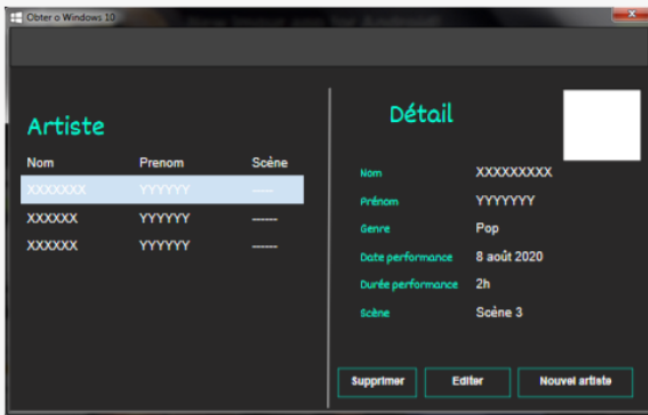


Figure 10 : Fenêtre gestion des artistes

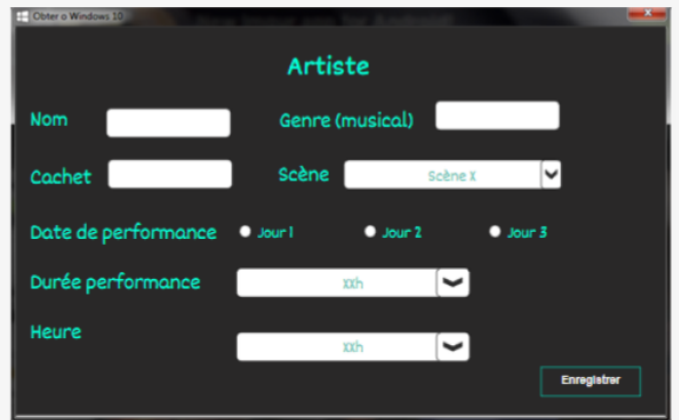


Figure 11 : Fenêtre artiste

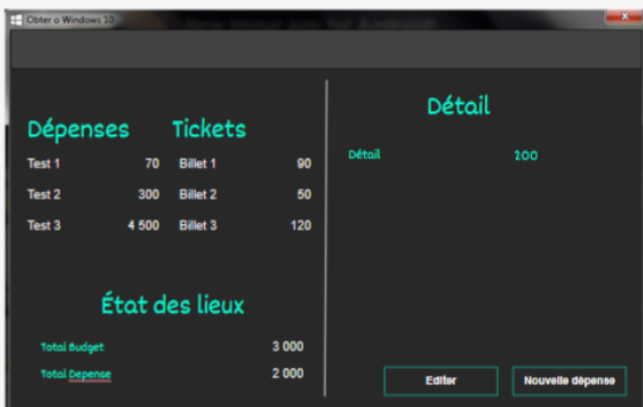


Figure 12 : Fenêtre gestion de budget

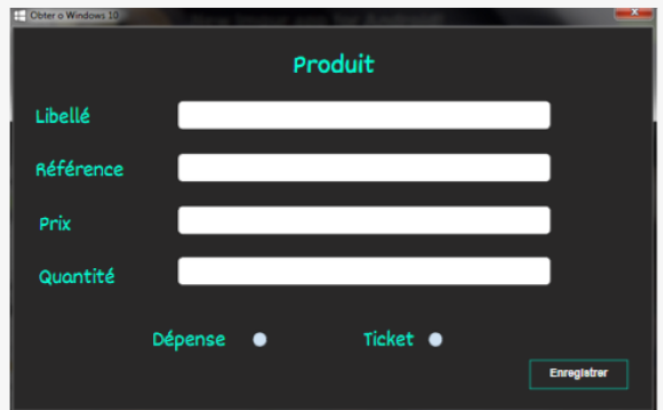


Figure 13 : Fenêtre produit

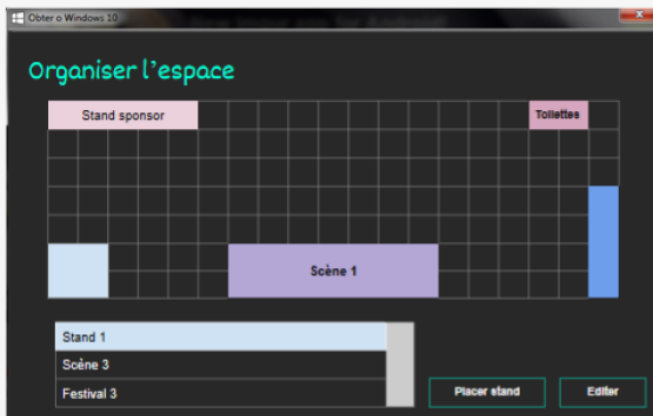


Figure 14 : Fenêtre Organiser Espace

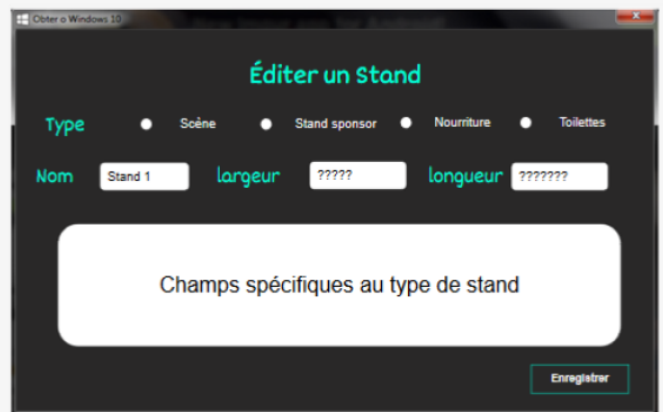


Figure 15 : Fenêtre éditer stands

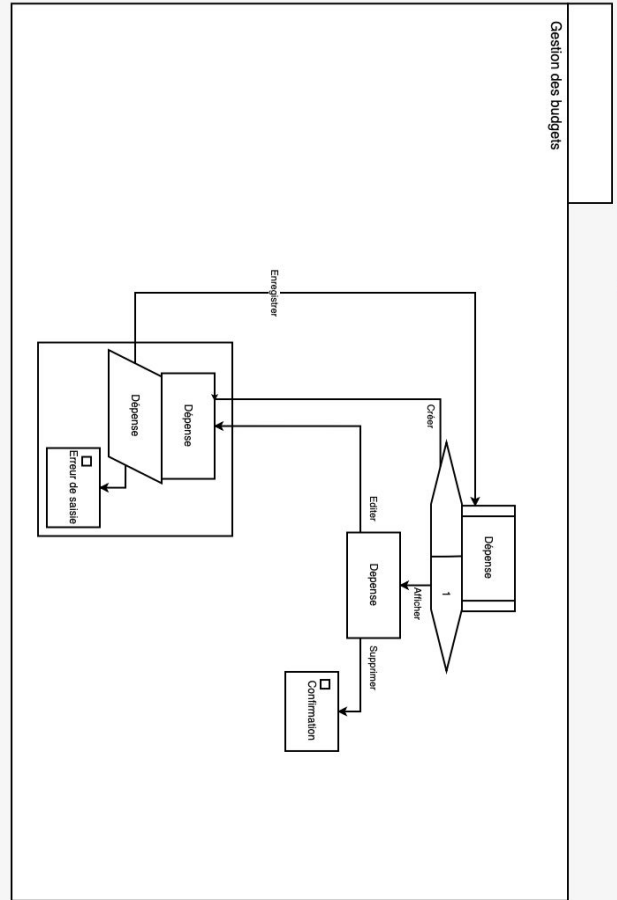


Figure 20 : SNI de l'écran de gestion du budget

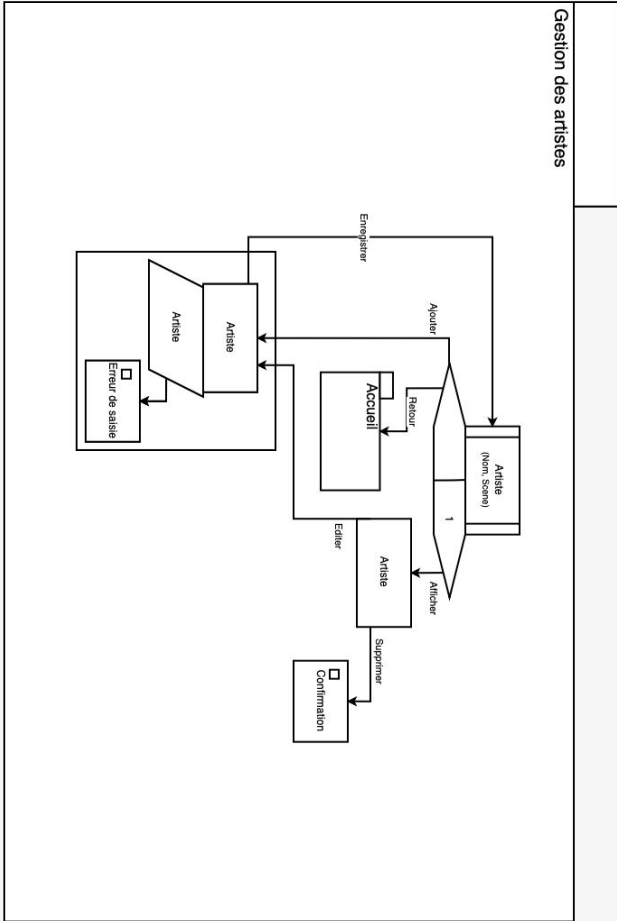


Figure 21 : SNI de l'écran de gestion des artistes

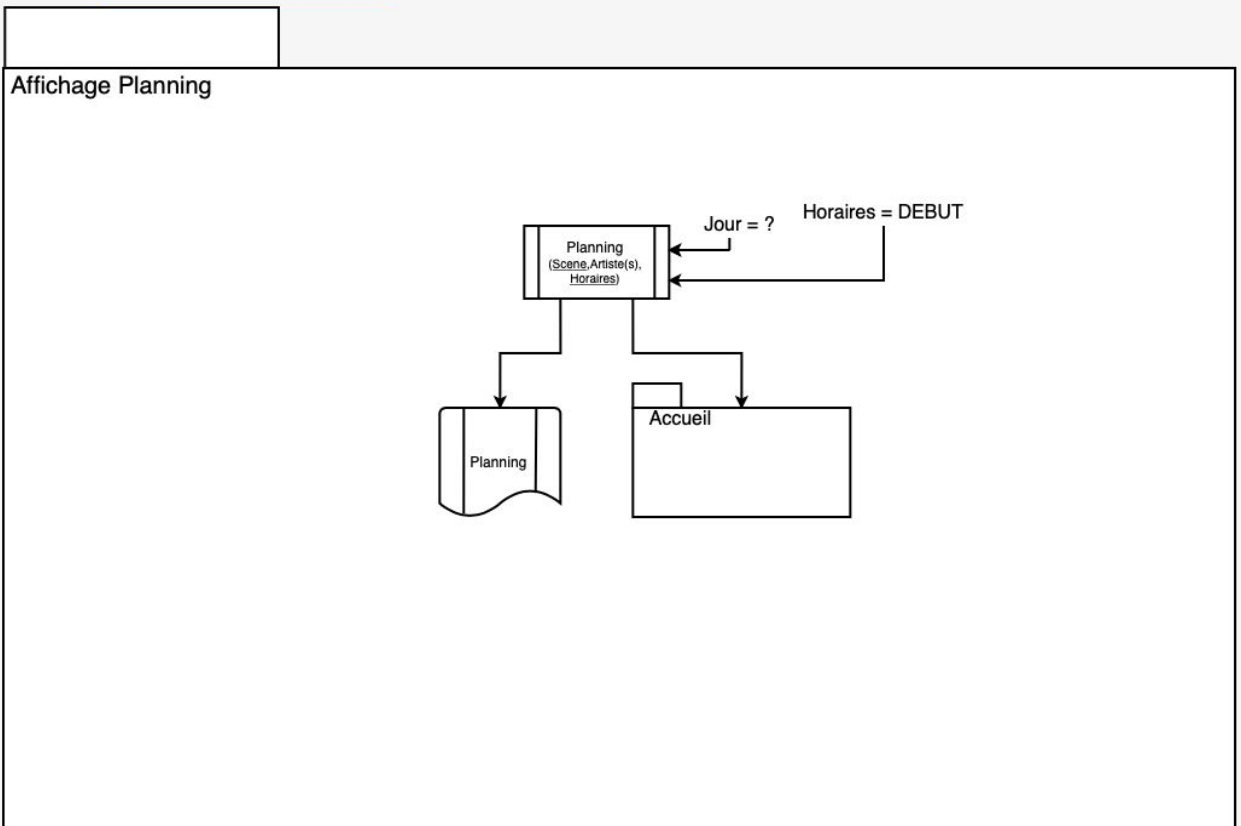


Figure 22 : SNI de l'écran d'affichage du planning

3.3. Diagramme de classes

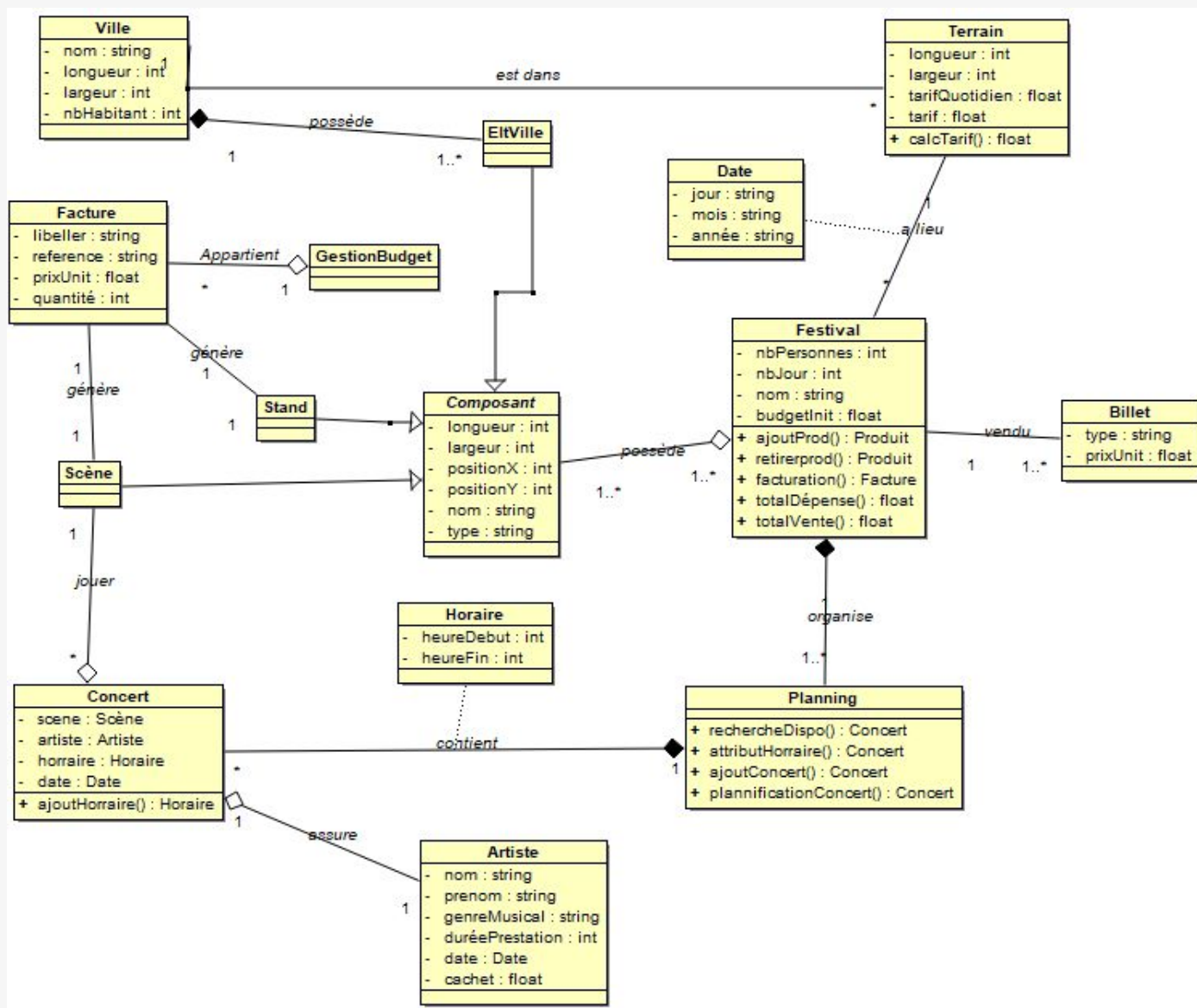


Figure 23 : Diagramme de classe

Dans un premier temps, l'utilisateur configure un festival en entrant toutes les données relatives à celui-ci : nom, nombre de personnes, nombre de jours, budget. Ainsi, on place également le festival dans une ville.

L'objectif premier de cette application reste de placer les composant du festival sur le site géographique. Ici, le logiciel peut gérer plusieurs type d'emplacements, comme pour des scènes, des stands. Tous ceux-ci font l'objet de facture que ce soit pour la location des scènes ou la vente d'emplacement pour les sponsors.

Ensuite, il est possible pour l'utilisateur de rentrer les informations concernant les artistes (nom, cachet, genre musicale) et les données concernant sa prestation (durée, date, scène). Une fois que l'artiste et le concert sont créés, celui-ci est ajouté au planning. Ainsi il est plus facile de visualiser la programmation du festival.

Le budget initial est demandé en paramètre du festival pour permettre au logiciel de calculer les dépenses et les ventes et ainsi de savoir quel est le profit du festival. Ceci est permis grâce à la génération de factures qui sont entrées dans le logiciel.

3.4. Explication des algorithmes

3.4.1. Algorithme 1 : Placement des éléments dans l'espace

But : Placer un composant (Stand, Scène) dans l'espace.

Principe:

- Générer un espace composé de carrés que l'on dispose successivement sur la scène pour créer un grand rectangle, l'équivalent de l'espace du festival, ce qui s'apparente à une grille.
- Il faut ensuite appliquer un listener sur chaque carré. Lors du clic sur le carré, l'attribut estTouché du carré est passé à vrai.
- On récupère le carré "Touché" et on lance les algorithmes de placement en fonction de la taille du composant.
- Puis, on recherche si la place est libre, c'est à dire qu'on pose une première fois le composant sur la scène (de manière invisible), se mettent alors en place 3 vérifications :
 1. Lors du placement il ne doit pas y avoir d'autres composants
 2. Le composant à placer ne doit pas empiéter sur un autre composant
 3. Le composant à placer ne doit pas déborder de l'espace du festivalAprès ces tests, on peut poser le composant sans encombre.
- La recherche se base sur la couleur. On connaît la couleur de la scène donc lors du parcours si la couleur du carré est différente c'est que la place est prise.

Le même algorithme est utilisé pour placer le festival dans la ville. Seulement, une fonction pour retourner le festival une fois qu'il est posé a été ajoutée. Pour la rotation, elle se fait si elle satisfait toutes les conditions des algorithmes précédents.

3.4.2. Algorithme 2 : Sérialisation des éléments

But : Sauvegarder les éléments importants (les dépenses/tickets, les artistes, les composants et les informations concernant le festival)

Principe:

- On déclare un nouveau package dans lequel on stockera nos éléments et qui possède une nouvelle classe (Liste). Cette dernière ne contient que des ArrayList d'objets que l'on veut sérialiser. Elle comprend aussi des méthodes pour ajouter les éléments dans les ArrayLists correspondantes et d'autres qui servent à retourner les ArrayList pour désérialiser.
- Ensuite, dans les classes où l'on ajoute les éléments, ces derniers seront ajoutés dans une instance de Liste déclarée au préalable. Après cela, nous allons sérialiser les informations en créant dans le package Main une classe sérialisation. Cette dernière contiendra une méthode qui sérialise la liste passée en paramètre. On déclare donc une instance de la classe sérialisation qu'on appelle après avoir ajouté les éléments dans la liste pour sérialiser

cette dernière. Après cela, nos éléments sont sérialisés et enregistrés dans un fichier.

- Vient ensuite l'extraction des données du fichier. Pour cela nous allons créer une méthode de désérialisation sans paramètre. Celle-ci va désérialiser la liste de la classe où elle se trouve. Après avoir désérialisé, nous avons une liste que nous devons exploiter pour remettre les éléments à leur place.
- Enfin, dans cette même méthode la liste désérialisée sera traitée en récupérant l'ArrayList spécifique à la classe pour récupérer les éléments un à un et les ajouter aux endroits correspondants.

Pour conclure, nous avons vu que la conception d'un logiciel est demande une grande réflexion, il faut imaginer toutes les fonctionnalités, les modélisées, penser à comment les mettre en place. Ce n'est pas un travail facile.

Nous pouvons maintenant passer à la concrétisation de tout cela et donc aborder le sujet de la réalisation, du développement du logiciel.

4. Réalisation

La réalisation est une étape importante pour la mise en oeuvre du projet. Elle permet la recherche et le développement de solutions pour satisfaire les objectifs définis. Le logiciel prend une forme concrète avec cette étape.

Dans cette partie, nous allons aborder les différentes fonctionnalités du logiciel avec une vue globale de notre projet.

Ici, nous allons, tout d'abord, mettre en lumière la documentation technique. Autrement dit, la maintenabilité de notre application.

Puis dans un dernier temps, nous allons présenter la documentation d'utilisation de logiciel.

4.1. Fonctionnalités

4.1.1. Placer le festival dans une ville

L'une des fonctionnalités principales de notre application est de placer un festival dans une ville. Nous avons donc programmé une grille sur laquelle se trouve des blocs représentant les magasins et les contraintes urbaines. On propose à l'utilisateur des emplacements avec des tarifs différents. Suivant la zone géographique, on peut orienter le festival. L'utilisateur peut alors placer ce dernier dans l'espace et changer son placement à sa guise pour trouver un emplacement plus intéressant.

4.1.2. Placer des éléments du festival sur le site géographique

Cette fonctionnalité reprend le principe de celle du dessus. Cependant, au lieu de placer le festival, l'utilisateur place les différents éléments du festival, c'est-à-dire les scènes, les stands qu'il a créé auparavant. Chaque ajout de composants génère une dépense supplémentaire. De plus on limite le nombre de scènes à trois.

4.1.3. Gestion des dépenses

Les dépenses et les ventes sont présentés à l'utilisateur dans deux TableView distincts. Dans ces derniers, nous pourrions retrouver le libellé et le prix de chaque produits vendus ou réalisés. Lorsque l'utilisateur sélectionne un produit dans l'un des deux TableView, les détails seront affichés sur la droite. L'utilisateur a la possibilité de saisir une nouvelle dépense ou une nouvelle vente puis remplir les champs nécessaires. Ce produit est donc ajouté au TableView selon sa catégorie. De plus, l'utilisateur a une vue sur l'état des lieux c'est-à-dire le budget total et les dépenses totales générés.

4.1.4. Gestion des artistes

Les artistes sont affichés dans un tableau TableView. Dans ce tableau on retrouve le nom de l'artiste, son genre musical et le cachet de sa prestation. Lorsque l'utilisateur sélectionne un artiste il peut afficher le détail de celui-ci. Tous les détails à propos de sa prestation vont donc être disponible, la date du concert, la durée de celui-ci et la scène sur laquelle il a lieu. Les prestations des artistes sont gérés directement depuis cet écran. Ainsi, lors de l'ajout d'un artiste, on ajoute sa prestation également. Cette classe envoie ensuite les informations au planning.

4.1.5. Affichage d'un planning

Un rectangle est créé et attribué pour chaque concert du festival. Ils sont ensuite affichés s'ils correspondent au jour sélectionné par l'utilisateur (premier jour par défaut). Avant d'être placés sur le planning selon leurs horaires, la scène qui leur est attribué, puis leurs dimensions sont ajustées selon leur durée. A chaque fois que l'utilisateur sélectionne un nouveau jour, les représentations de concerts disparaissent avant que celles du nouveau jour apparaissent selon ce même mode opératoire.

4.2. Documentation utilisateur

Cette documentation permet à l'utilisateur de l'aide lors de la première session sur l'application. Pour les sessions à venir, cette documentation possède une partie entière concernant les erreurs que l'application peut générer. Ainsi, l'utilisateur peut être autonome et indépendant pour la résolution de problèmes. De plus, la documentation utilisateur indique toutes les manipulations à faire pour chaque module de l'application pour utiliser celle-ci au mieux. Chaque manoeuvre est expliquée avec des schémas et des phrases simples pour que quiconque lise la documentation puisse utiliser l'application.

4.3. Documentation technique

Cette documentation permet d'aider les développeurs pour mieux comprendre le code. Pour chaque classe on donne l'auteur et on commente les méthodes les plus intéressantes. La doc technique sert surtout lorsque les collaborateurs veulent comprendre le code ou pour ajouter des fonctionnalités ultérieurement . (cf dossier doc sur le gitlab)

Pour conclure, nous avons vu que la réalisation d'un logiciel est une étape longue et que le souci du détail est important. Il faut prendre en compte que le logiciel n'est peut-être pas évident pour l'utilisateur.

Nous allons, à présent, passer aux résultats obtenus et reprendre l'ensemble de notre travail effectué.

5. Conclusion

Notre projet n'est pas la meilleure version de lui-même. Néanmoins, avec les contraintes de temps et d'environnement, nous avons fait de notre mieux pour réaliser un projet plus qu'acceptable. Comme disait *Marcel Proust*: "*Il n'y a pas de réussite facile, ni d'échecs définitifs*".

Hormis cela, nous ne nous étions pas rendu compte, au début, de l'ampleur que prend un projet. Les contraintes, les risques, le retard, la mauvaise gestion des lots etc. nous ont fait prendre conscience de la complexité d'un tel travail. Chaque tâche finie en révélait une nouvelle plus rude. Il était nécessaire de mettre en éveil notre réflexion, notre sens de coopération et d'entraide pour aboutir la mission.

De plus, une partie de nos objectifs ont été atteints. Voici quelques exemples. L'ergonomie de notre logiciel. Ce dernier est simple, complet et est efficace au niveau de l'affichage. L'utilisateur n'est pas égaré parmi toutes les fonctionnalités proposées par l'application. En outre, cette dernière est performante c'est-à-dire que sa rapidité permet à l'utilisateur de ne pas se décrocher. La gestion des ventes est très fonctionnelle aussi. L'utilisateur peut consulter les recettes générées au cours des jours.

Néanmoins, malgré ces objectifs qui ont été atteints grâce à notre équipe, d'autres n'ont pas réussi à arriver à leur but. Par exemple, le temps prévu dans le diagramme de Gantt n'a pas été respecté. La majorité de nos tâches ont prit plus de délai comme nous l'avions dit précédemment. D'ailleurs, la répartition de certains lots n'a pas été prise en compte le long de la conception et réalisation de notre projet.

Malgré ces quelques difficultés rencontrées, ce travail en groupe nous a permis d'enrichir nos connaissances dans plusieurs domaines de l'informatique et d'élargir nos compétences. Nous avons pris conscience du processus d'un projet, de son acheminement de A à Z. Nous avons concilié notre motivation et notre esprit d'équipe pour mener à bien ce travail relativement compliqué. Nous avons fait preuve de flexibilité et de détermination tout au long du projet.

Table des illustrations

Figure 1 : Bête à cornes	3
Figure 2 : Diagramme des tâches	5
Figure 3 : Diagramme des responsabilités	6
Figure 4 : Matrice RACI	6
Figure 5 : Diagramme de Gantt	7
Figure 6 : Écran d'accueil de l'application Festi'GO	8
Figure 7 : Écran de création d'un festival	8
Figure 8 : Écran d'affichage des plannings	8
Figure 9 : Écran pour poser le festival	8
Figure 10 : Fenêtre gestion des artistes	9
Figure 11 : Fenêtre artiste	9
Figure 12 : Fenêtre gestion de budget	9
Figure 13 : Fenêtre produit	9
Figure 14 : Fenêtre organiser l'espace	9
Figure 15 : Fenêtre éditer stand	9
Figure 16 : SNI de l'écran d'accueil	10
Figure 17 : SNI de l'écran de paramétrage du festival	10
Figure 18 : SNI de l'écran du placement du festival	10
Figure 19 : SNI de l'écran de l'organisation	10
Figure 20 : SNI de l'écran de gestion du budget	11
Figure 21 : SNI de l'écran de gestion des artistes	11
Figure 22 : SNI de l'écran de l'affichage des plannings	11
Figure 23 : Diagramme de classe	12

Annexe

Bibliographie

- Jenkov, Jacob. (2020, 25 février). JavaFX TableView sur <http://tutorials.jenkov.com/javafx/tableview.html>
- Sample TableView avec 2 colonnes, RIP Tutorial ici <https://riptutorial.com/fr/javafx/example/7288/sample-tableview-avec-2-colonnes>
- TableView, (2016, 2 novembre) Developpez.com sur <https://java.developpez.com/faq/javafx?page=TableView>
- Le tutoriel de JavaFX TableView, O7planning ici <https://o7planning.org/fr/11079/tutoriel-javafx-tableview>
- Javadoc - TableColumn sur <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TableColumn.html>
- Manu, Remy, Cours IHM-1 JavaFX, Conteneurs - Layout Panes, 64 pages, disponible ici http://remy-manu.no-ip.biz/Java/Tutoriels/JavaFX/PDF/ihm1_fx_04_man.pdf
- Java ArrayList, w3schools.com disponible ici https://www.w3schools.com/java/java_arraylist.asp
- Manu, Remy, Cours IHM-1 JavaFX, Architecture MVC - Gestion des évènements, 50 pages, disponible ici http://remy-manu.no-ip.biz/Java/Tutoriels/JavaFX/PDF/ihm1_fx_06_man.pdf
- Boutons, (2016, 2 novembre) Developpez.com sur <https://java.developpez.com/faq/javafx?page=Boutons>
- Jakob, Marco. (2015, 15 avril). Partie 4 : Style CSS sur <https://code.makery.ch/fr/library/javafx-tutorial/part4/>
- José Paumard, Sérialisation d'objets - Enjeu de la sérialisation d'objets disponible ici <http://blog.paumard.org/cours/java/chap10-entrees-sorties-serialization.html>
- GeeksProgramming - Sérialisation et désérialisation : Java sur <https://www.youtube.com/watch?v=4pJOy9Dv9q0>
- O7planning - Tutoriels de programmation JavaFX sur <https://o7planning.org/fr/11009/javafx>
- La sérialisation, (2016, 19 mars) Developpez.com sur <https://jmdoudoux.developpez.com/cours/developpons/java/chap-serialisation.php>
- OpenClassroom le cours sur javafx sur <https://openclassrooms.com/fr/courses/26832-apprenez-a-programmer-en-java/5014486-decouvrez-java-fx>
- Javadoc sur <https://docs.oracle.com/javase/8/javafx/api/>
- Les Teachers du Net - Apprendre JavaFX disponible sur https://www.youtube.com/watch?v=02SFViTyX4A&list=PLlxQJeQRaKDSIq_sOFKqBHvFsox2ZVWbs

- Les Teachers du Net - Comment sérialiser un objet en Java ? disponible sur <https://www.youtube.com/watch?v=xzXEgUzRFh8>
- Les Teachers du Net - Comment désérialiser un objet en Java ? disponible sur <https://www.youtube.com/watch?v=5jDY0H5ITZo>

« Préface »

Cette application est destinée aux organisateurs du festival. En aucun cas celle-ci ne doit être utilisée par autrui, l'application gérant certaines données sensibles (la gérance des factures et des dépenses de l'utilisateur) ne doit être vue de tous. De plus, l'application ne marche que en local, ceci implique de toujours utiliser l'application sur le même ordinateur.

Cette application a pour but d'aider l'utilisateur dans la gestion du festival sur certains aspects.

Ce document est là pour vous aider à prendre en main le logiciel et vous apprendre les manipulations à faire en cas d'erreur. Il permettra à l'utilisateur de se servir au mieux de l'application et de profiter au maximum de ses spécificités. Il a également pour but d'aider l'utilisateur à utiliser l'application pour le soulager de la charge de travail en laissant celle-ci à l'application.

Sommaire

« Préface »	0
Sommaire	1
I- La première mise en route	2
Liste des modules	2
II-Gestion du festival	3
Paramètre festival	3
→ Création d'un festival	3
Placer festival	4
→ Placer le festival	4
Organiser festival	5
→ Gérer les stands	5
→ Création des stands	5
→ Placer les stands	5
III-Gestion des Artistes et du planning	6
Afficher planning	6
→ Création des plannings	7
Gestion des artistes	7
IV- Gestion des dépenses	9
Ajout d'une dépense	9
Ajout d'une vente	11
V-Gestion des erreurs	14
Les erreurs de saisie	14
Matrice de saisie pour la fenêtre festival:	14
Matrice de saisie pour la fenêtre de création d'un produit :	16
Matrice de saisie pour la fenêtre de création d'un artiste :	17
Matrice de saisie pour la fenêtre de création d'un stand :	18
Contact des développeurs :	20

I- La première mise en route

Ce premier chapitre sera une sorte d'introduction de l'application. Nous vous ferons une liste des modules disponibles dans l'application et les différents usages que l'on peut en faire. En effet, cette application vous facilitera la gérance de votre festival mais celle-ci n'est pas une solution miracle et possède ses limites.

1. Liste des modules

Lorsque vous lancerez l'application vous tomberez toujours sur la même fenêtre, même si vous avez déjà créé un festival. Vous devriez donc normalement tomber sur cette fenêtre :

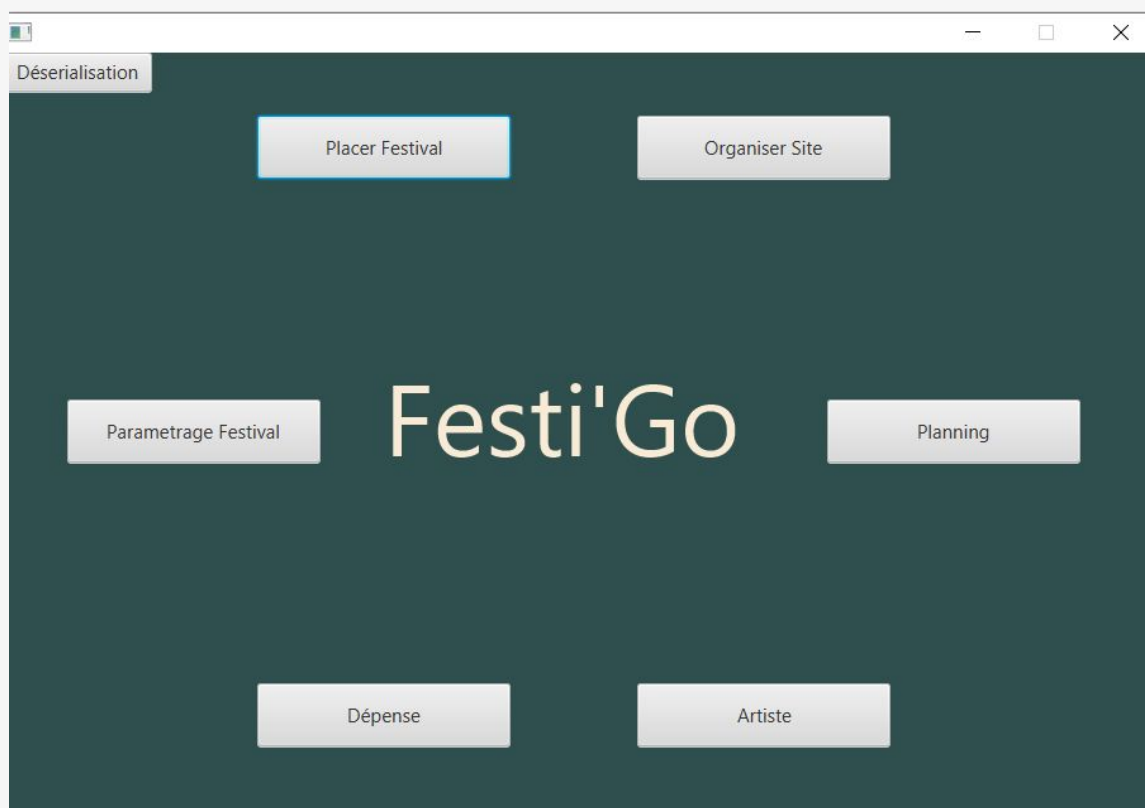


Figure 1 : Écran d'accueil de l'application de Festi'GO

Avant de vous précipiter et de faire une erreur malencontreuse sur l'application, nous allons d'abord faire un tour des différents modules qui existent et expliquer les grandes lignes de leurs fonctions. Nous verrons plus loin dans ce chapitre un fonctionnement un peu plus approfondi des modules.

Le 1^{er} module est « Paramétrer festival ». Il se trouve à gauche de la fenêtre. Pour la suite des modules nous irons dans le sens des aiguilles d'une montre.

Ce premier module est donc le premier que vous utiliserez car il permet de créer votre festival et de l'enregistrer.

Le deuxième module, « Placer festival » est le second que vous utiliserez car après avoir créé votre festival il vous faudra le placer. Cela peut vous paraître étrange, mais vous verrez plus tard comment procéder.

Le troisième module, « Organiser Espace » permettra, au sein de votre festival, de placer vos éléments (tels que les scènes, les stands, les toilettes,...). Il sera le troisième module que vous utiliserez après les deux premiers.

Le quatrième module, « Afficher planning » porte son nom et vous affichera tout simplement le planning des concerts qui ont été créés.

Le cinquième module, « Gestion budget » permettra à vous, utilisateurs, d'entrer vos dépenses et bénéfices durant le festival pour ainsi voir si le festival a permis de gagner de l'argent.

Enfin le dernier module, « Gestion Artistes », ce module va vous permettre de créer vos artistes qui joueront durant le festival et ainsi créer les concerts qui apparaîtront dans le quatrième module.

II-Gestion du festival

1. Paramètre festival

Dans ce module, vous apprendrez donc à créer un festival. Pour cela, vous tomberez sur une fenêtre dans lequel vous devrez remplir des champs pour ensuite enregistrer le festival. L'explication des champs sera fait ultérieurement avec une explication plus précise.

→ Création d'un festival

Pour créer un festival s'il n'existe pas déjà, vous devez cliquer sur le bouton "paramètre festival".

Il ouvre une fenêtre dans laquelle vous pouvez entrer les différents paramètres :

- Le nom
- Le budget total (un nombre entier)
- Le nombre de places maximum du festival (un nombre entier)
- La durée du festival (en jours et 3 maximum)
- La date de début du festival
- Le noms des organisateurs
- Le nombre maximal d'artistes prévu

Une fois toutes ces informations entrées, pour enregistrer vous n'avez plus qu'à cliquer sur le bouton prévue à cet effet.

Figure 2 : Écran paramètre festival

a. Placer festival

Dans ce module, vous devrez placer votre festival dans une simulation d'une ville générée aléatoirement. Ceci permettra de prendre en compte la distance entre votre festival et la ville et/ou de placer le festival selon les directives de la municipalité.

→ Placer le festival

Afin de placer le festival, depuis la fenêtre d'accueil vous devez cliquer sur le bouton "placer festival".

Cette fenêtre génère automatiquement un "terrain" (ou espace) dans une ville avec des bâtiments placés aléatoirement.

Si vous n'êtes pas satisfaits, vous pouvez en générer un nouveau en cliquant sur le bouton "Changer emplacement".

Une fois que le terrain vous convient, vous n'avez qu'à placer le festival en cliquant sur deux cases qui délimitent votre emplacement de festival.

A savoir que votre festival prendra une forme rectangulaire avec comme sommets opposés les deux cases définies.

Quand l'emplacement vous convient vous n'avez plus qu'à cliquer sur le bouton "enregistrer".

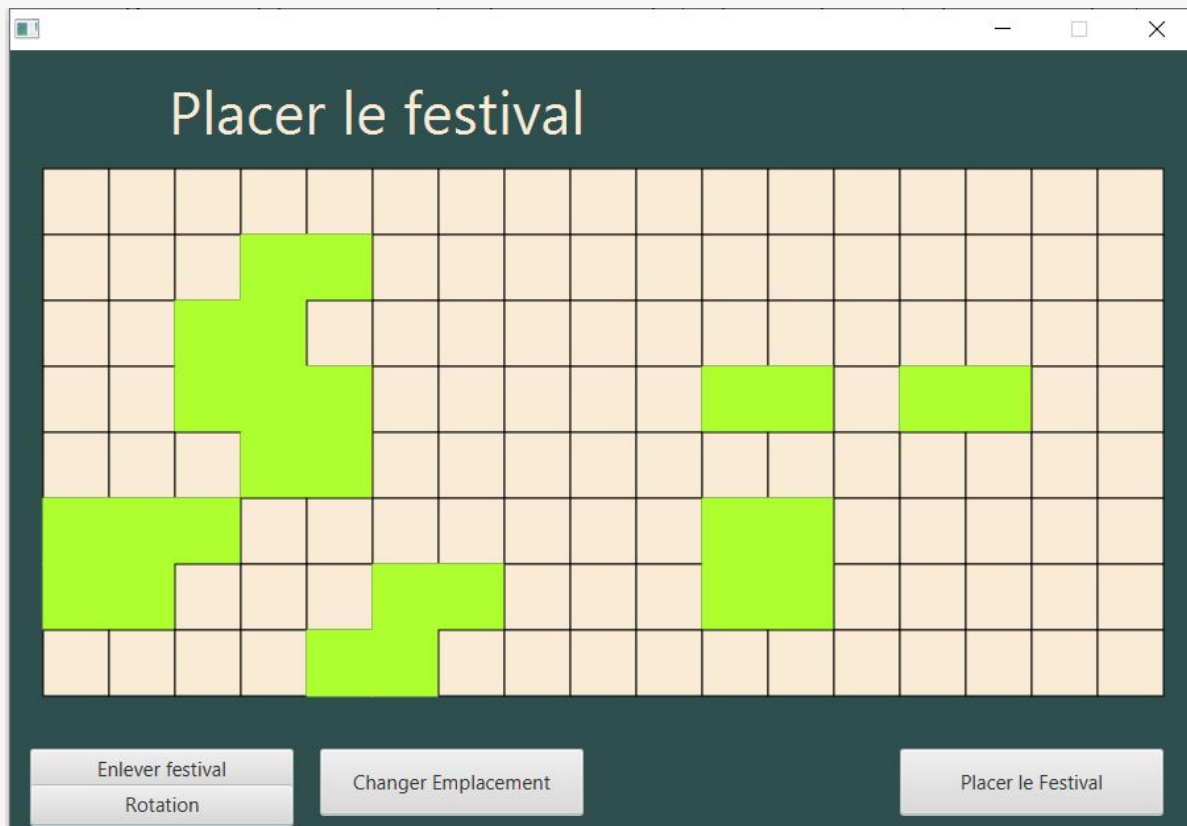


Figure 3 : Écran poser festival

b. Organiser festival

Dans ce module, après avoir placé votre festival, vous serez amené à aménager votre festival. Ceci se fera en créant les éléments nécessaires que vous aurez besoin. Pour créer un élément, vous serez envoyé sur la page correspondante où vous devrez remplir les informations concernant le stand.

→ Gérer les stands

Pour placer les éléments du festival, vous devez cliquer sur le bouton "Organiser espace"

→ Création des stands

Avant de placer les stands, vous devrez les créer en sélectionnant "?Nouveau Stand?".

Sur la fenêtre ouverte, vous devrez alors entrer/sélectionner les informations correspondantes au stand que vous voulez créer.

Une fois ceci effectué, vous n'aurez plus qu'à valider.

→ Placer les stands

Une fois les stands créés, pour les placer dans le festival, vous devez les sélectionner dans la liste de droite puis cliquer sur la case où vous souhaitez le placer.

Si la direction du stand ne vous convient pas, vous pouvez le faire pivoter grâce au bouton "Rotation".

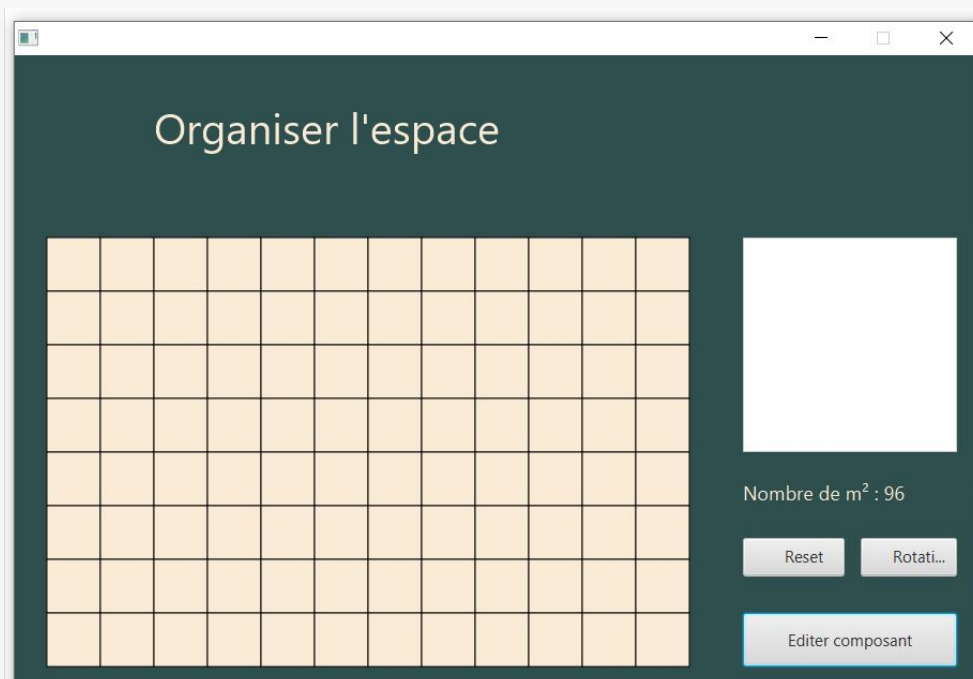


Figure 4 : Écran organiser l'espace

Figure 5 : Écran de création d'un stand

III-Gestion des Artistes et du planning

c. Afficher planning

Ce module vous permettra à la suite de création d'artiste et de ses concerts de voir ses performances sur un planning qui se remplira automatiquement selon les représentations créées.

→ Création des plannings

Pour accéder au planning vous pouvez, à partir de la page d'accueil, cliquer sur le bouton "Afficher planning".

Vous pourrez ensuite consulter le planning en fonction du jour si votre festival se déroule sur plusieurs journées.

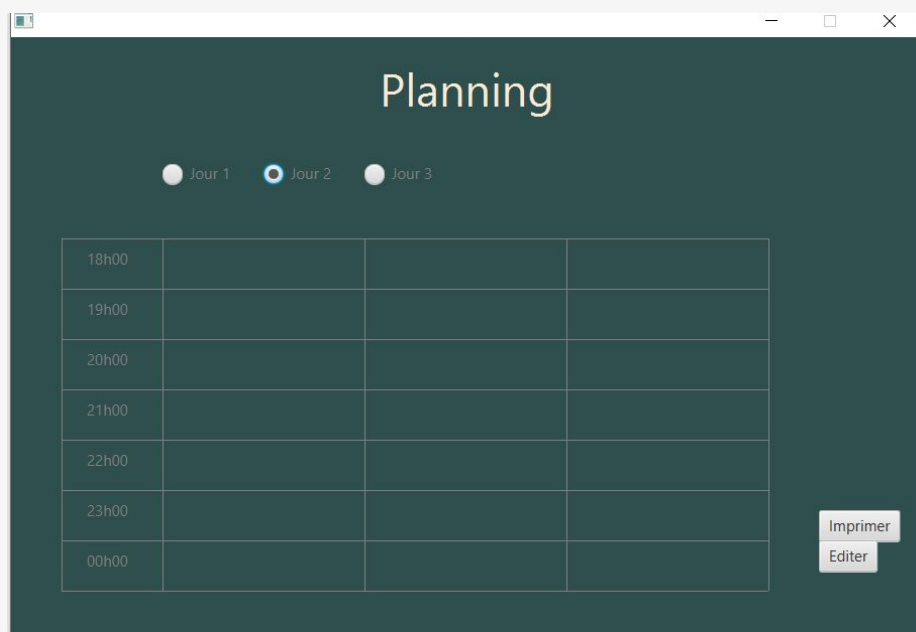


Figure 6 : Exemple de planning

d. Gestion des artistes

Ce module va vous permettre de créer les artistes qui vont jouer durant votre festival. Durant la création d'un artiste, vous devrez indiquer le jour de passage et l'heure pour ainsi créer le concert et le placer dans le planning.

→ Ajout d'un artiste

Pour entrer un artiste dans le programme, il faut cliquer sur le bouton "Gestion artistes" puis "Nouvel artiste". Un fois sur la fenêtre vous devrez remplir tu les champs:

-Nom

-Genre musical

-Cachet

-Scène

-Date de performance

-Durée de performance en heure

-Heure de la performance

→ Edition d'un artiste

Pour éditer un artiste, il vous faudra sélectionner celui que vous souhaitez dans la liste de gauche puis cliquer sur "Editer", vous vous retrouvez alors dans le même cas qu'un création.

→ Suppression d'un artiste

Pour supprimer un artiste, il vous faudra sélectionner celui que vous souhaitez dans la liste de gauche puis cliquer sur "Supprimer".

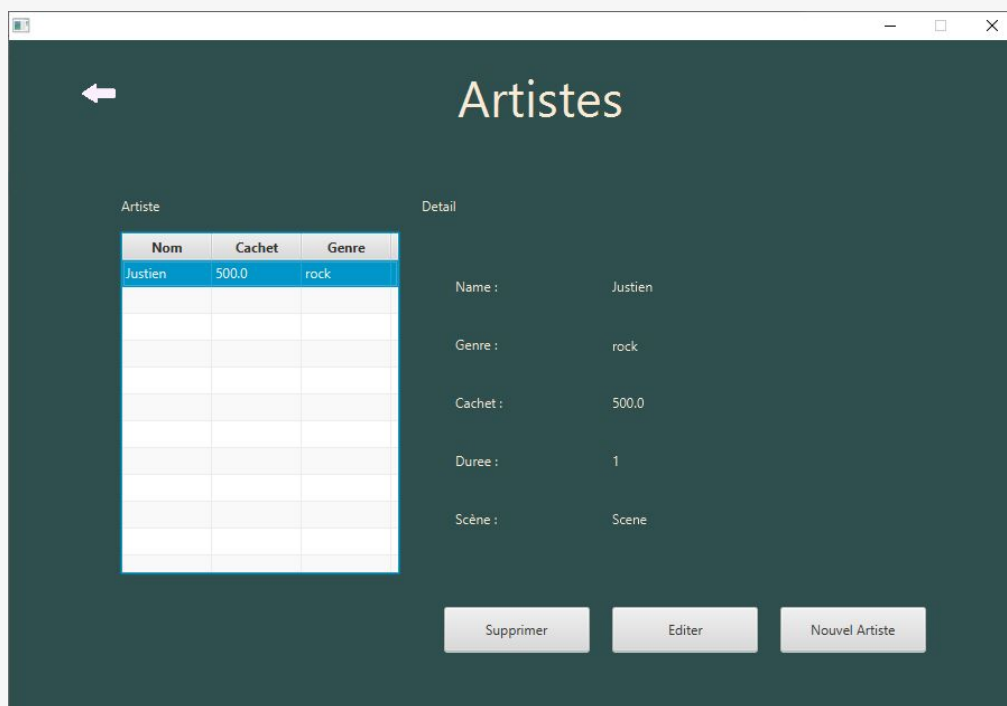


Figure 7 : Écran des artistes

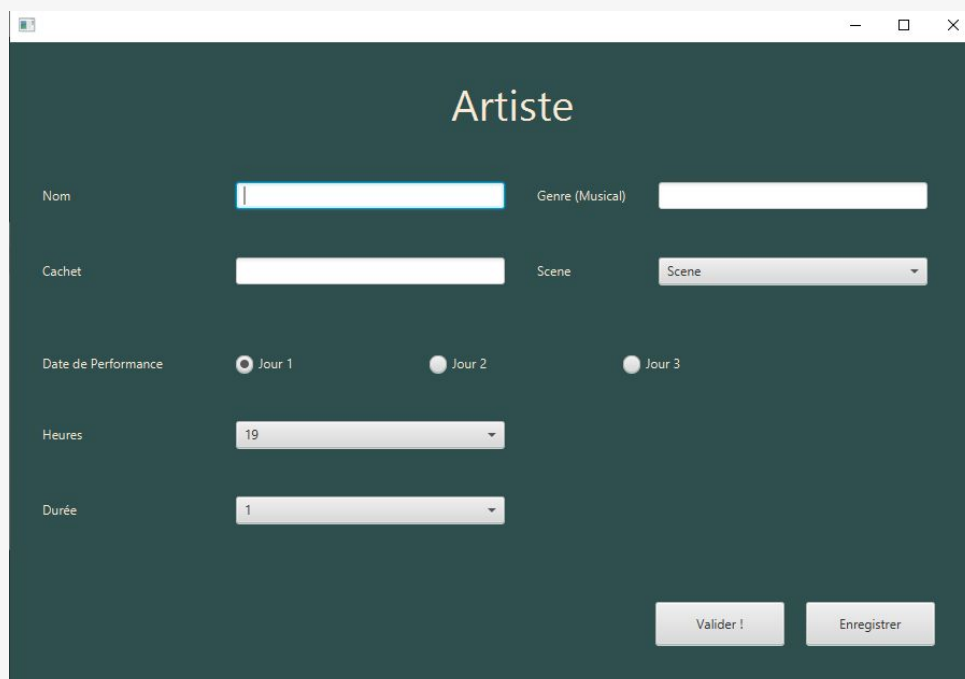


Figure 8: Écran de création d'un artistes

IV- Gestion des dépenses

Dans cette partie, vous allez apprendre à laisser l'application gérer le festival. En effet le but d'un festival est aussi de récupérer des bénéfices pour payer les gens. Mais nous sommes d'accord que faire tous les calculs à la main peuvent être long et fastidieux. De plus, des erreurs de calculs peuvent se produire et lors d'une communication à ses supérieurs on peut donner des mauvais chiffres. Notre application ne peut palier aux erreurs de communications. Cependant, elle peut vous aider à calculer vos dépenses et bénéfices et voir si oui ou non vous avez réalisé un bon festival.

1. Ajout d'une dépense

Dans cette première partie, vous allez apprendre à ajouter une dépenses. Les dépenses sont les cachets données au artistes, les dépenses liées au scènes, au stand... Pour cela il faut d'abords remplir quelques conditions avant de pouvoir ajouter des dépenses :

- Il faut avoir créer un festival
- Il faut avoir placé le festival
- Il faut également avoir placé les éléments/stands de votre festival

Si ces conditions sont réunies, alors nous allons pouvoir commencer à créer des dépenses. Cliquez sur le module entourer en rouge.

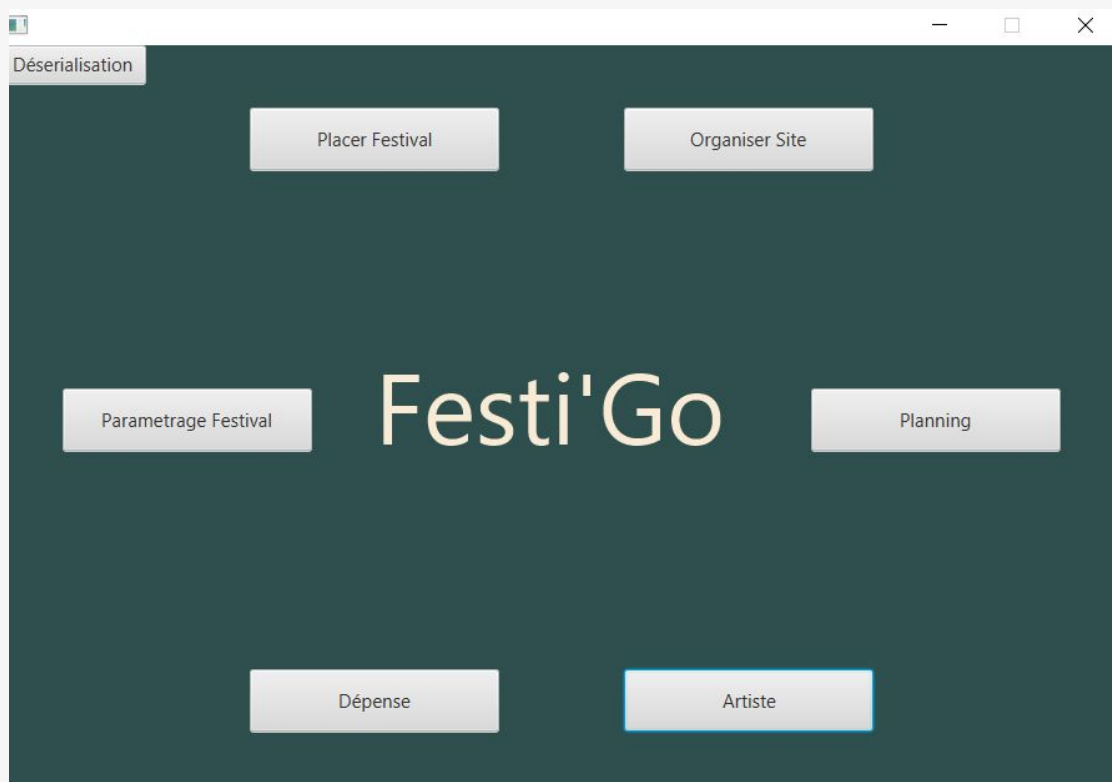


Figure 9: Écran d'accueil de l'application de festi'GO

Après cela, vous devriez normalement tomber sur cette fenêtre ci-dessous. Si ce n'est pas le cas, alors fermer l'application et relancer là. Et si ce problème persiste rendez vous à la page XX (partie 5) avec la gestion des erreurs.

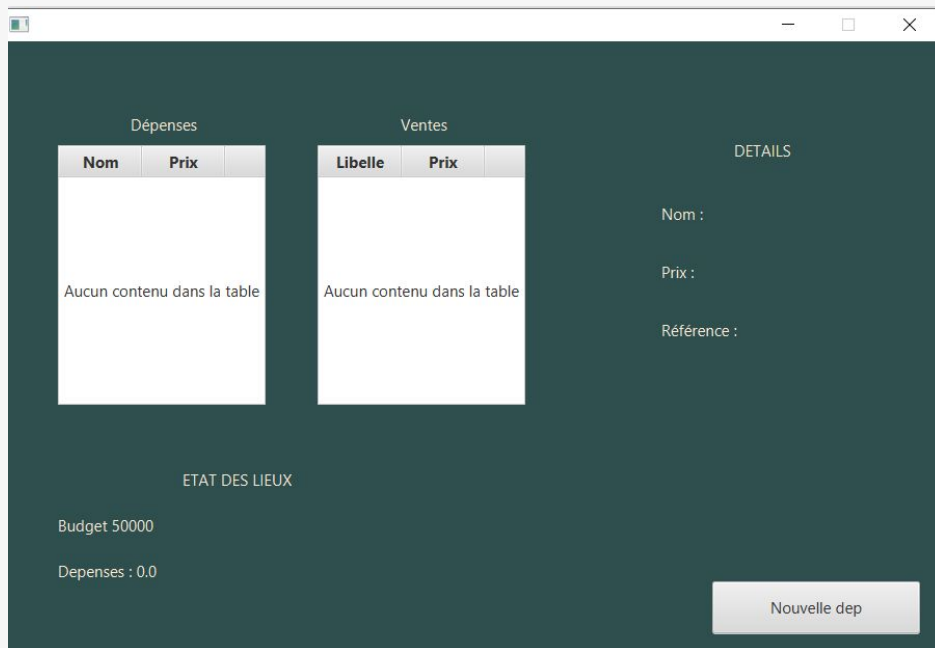


Figure 10 : Écran des dépenses et des bénéfices

Vous allez trouver sur cet écran le budget initial que vous avez entré lors du paramétrage du festival (cercle rouge). Ainsi que deux listes, une pour les dépenses et une pour les ventes (bénéfices). On a aussi une zone détails que l'on verra plus tard son utilité.

Pour créer une nouvelle dépense, appuyez sur le bouton « nouvelle dep » (le bouton entouré en vert). A la suite de ce clic, vous allez arriver sur une nouvelle fenêtre (voir ci-dessous). Si un problème arrive, redirigez vous à la page XX ou relancez l'application.

The 'Produit' form is used to create a new expense entry. It contains the following fields and options:

- Libellé:** Text input field containing 'Frites'.
- Quantité:** Text input field containing '12'.
- Référence:** Text input field containing 'aze1'.
- Prix:** Text input field containing '10'.
- Category:** Two radio buttons: 'Ticket' (unselected) and 'Dépense' (selected).
- Buttons:** 'Validez !' (Validate) and 'Enregistrer' (Save) buttons at the bottom.

Figure 11 : Écran de création d'une dépenses

Voici un exemple de comment remplir la fenêtre. Le 1^{er} champ correspond au nom du produit ou de la dépense. Ensuite, il vous faut entrer une quantité, si vous n'avez aucune quantité (un cachet par exemple) mettez 1 comme quantité. Cela permettra d'éviter les erreurs et d'utiliser avec moins d'erreurs l'application. Ensuite, il vous faut saisir la référence de votre produit, ici c'est un exemple mais seul vous connaissez ce

champs à remplir . Ensuite, remplissez le prix unitaire de votre article et enfin choisissez si votre produit est une dépense ou une vente (Ticket), dans notre cas présent il vous faut cliquer sur le rond à côté de dépenses comme sur l'exemple ci-dessus. Pour que votre achat soit enregistré, cliquez d'abord sur « Validez ! » et enfin sur « Enregistrer ». Après ce dernier clic la fenêtre se fermera automatiquement pas de panic et vous serez renvoyé sur la fenêtre précédente. Cependant, cette fenêtre aura subi des changements puisque vous allez pouvoir voir que votre dépenses a été ajoutée dans la liste de gauche (voir ci-dessous).

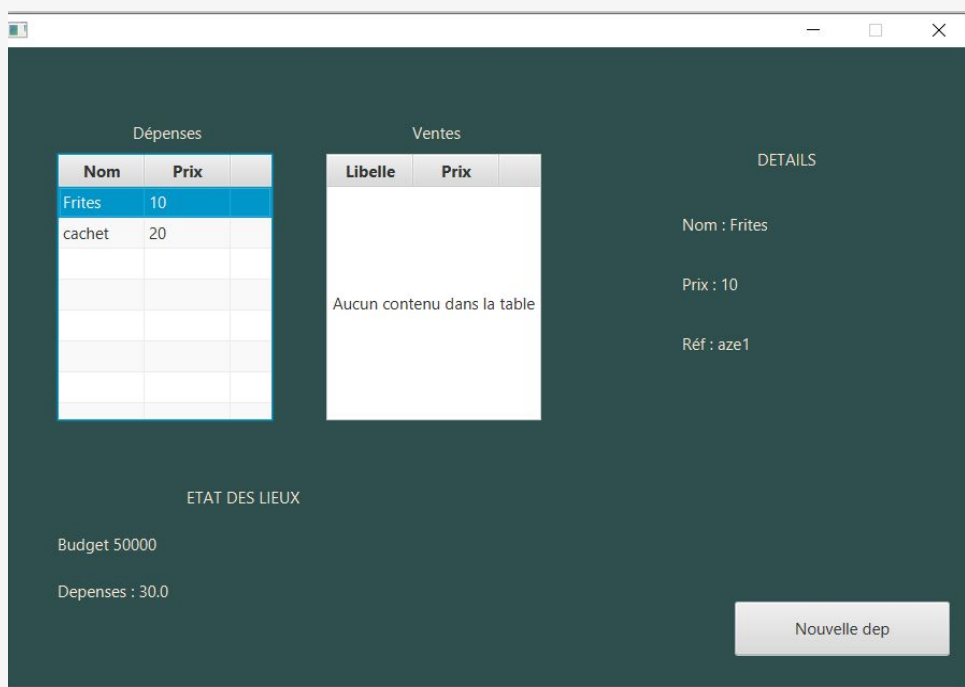


Figure 12 : Écran affichage des dépenses et des bénéfice

Vous pouvez voir que le nombre de dépenses à changer puisque dans notre exemple celui-ci est passé à 30 qui est la somme de 10 x la quantité et 20 x la quantité. De plus, lorsque vous cliquez sur une dépense vous pouvez voir les détails de cette dépense (cercle bleu).

Vous savez maintenant ajouter une dépense, vous pouvez tester et créer vos dépenses du moment qu'elle reste raisonnable et en dessous du budget sinon une erreur se produira.

2. Ajout d'une vente

Dans cette deuxième partie vous allez apprendre à ajouter une vente. Les ventes représentent principalement les tickets vendus pour le festival ainsi des produits dérivés vendus (boisson,t-shirt...). Pour cela, il faut d'abord remplir quelques conditions avant de pouvoir ajouter des ventes :

- Il faut avoir créé un festival
- Il faut avoir placé le festival
- Il faut également avoir placé les éléments/stands de votre festival

Si ces conditions sont réunies, alors nous allons pouvoir commencer à créer des ventes. Cliquez sur le module entouré en rouge.

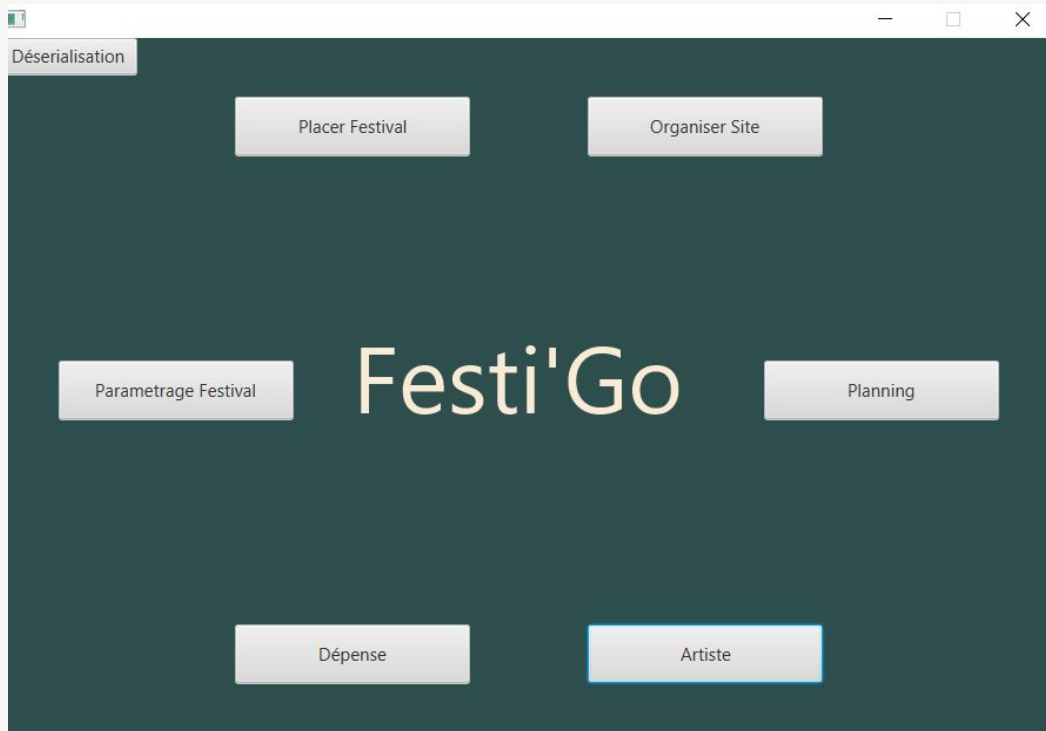


Figure 13 : Écran d'accueil de l'application de festi'GO

Après cela, vous devriez normalement tomber sur cette fenêtre ci-dessous, si ce n'est pas le cas alors fermez l'application et relancez là. Et si ce problème persiste, rendez vous à la page XX (partie 5) avec la gestion des erreurs.

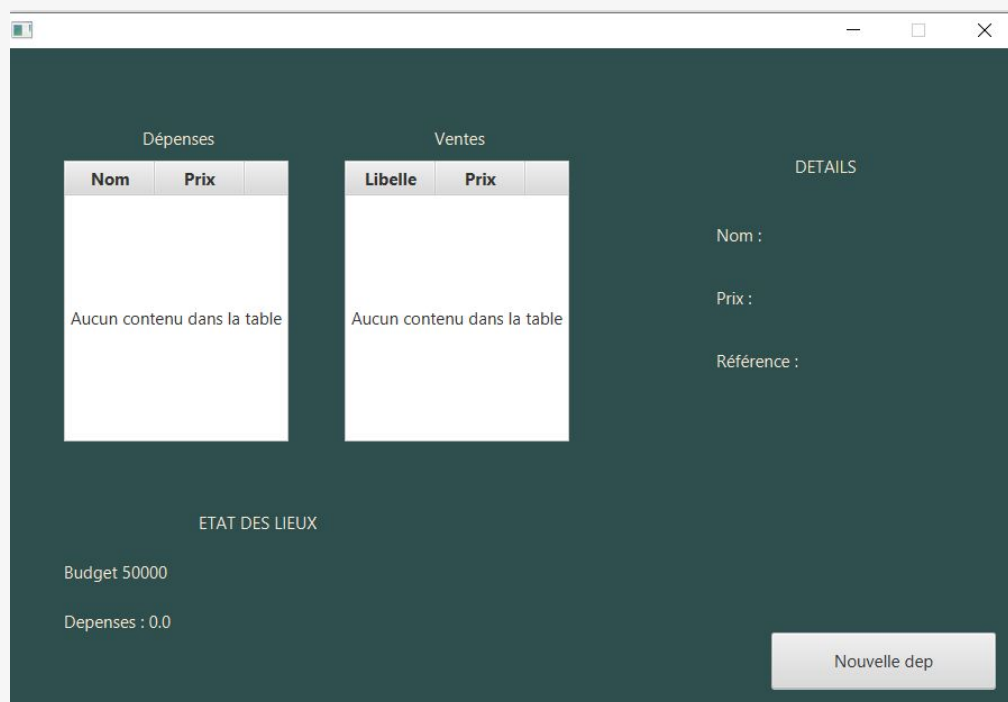


Figure 14 : Écran des des dépenses et des ventes

Vous allez trouver sur cet écran le budget initial que vous avez entré lors du paramétrage du festival (cercle rouge). Ainsi que deux listes, une pour les dépenses et une pour les ventes (bénéfices). On a aussi une zone détails que l'on verra plus tard son utilité.

Pour créer une nouvelle vente, appuyez sur le bouton « nouvelle dep » (le bouton entouré en vert). A la suite de ce clic vous allez arriver sur une nouvel fenêtre (voir ci-dessous). Si un problèmes arrive, dirigez vous à la page 17 ou relancer l'application.

Produit

Libellé ticket Adulte

Quantité 10

Référence TAZ1

Prix 10

Ticket Dépense

Validez ! Enregistrer

Figure 15 : Écran de création d'une facture

Voici un exemple de comment remplir la fenêtre. Le 1^{er} champs correspond au nom du produit ou de la vente. Ensuite, il vous faut entrer une quantité , si vous n'avez aucune quantité (un cachet par exemple) mettez 1 comme quantité, cela permettra d'éviter les erreurs et d'utiliser avec moins d'erreur l'application. Ensuite il vous faut taper la références de votre produit, ici c'est un exemple mais seul vous, connaissez ce champs à remplir . Ensuite remplissez le prix unitaire de votre article et enfin choisissez si votre produit est une dépense ou une vente (Ticket), dans notre cas présent il vous faut cliquer sur le rond à côté de tickets comme sur l'exemple ci-dessus.Pour que votre achat soit enregistré, cliquez d'abord sur « Validez ! » et enfin sur « Enregistrer ». Après ce dernier clic, la fenêtre se fermera automatiquement. Ne paniquez pas, vous serez renvoyés sur la fenêtre précédente. Cependant cette dernière aura subi des changements puisque vous allez pouvoir voir que votre vente à été ajoutée dans la liste de droite (voir ci-dessous).

Dépenses		Ventes	
Nom	Prix	Libelle	Prix
Frite	10	ticket Ad...	10
Cachet	20		

DETAILS

Nom : ticket Adulte

Prix : 10

Réf : TAZ1

ETAT DES LIEUX

Budget 50000

Depenses : 40.0

Nouvelle dep

Figure 16 : Écran des dépenses et ventes après ajouts

Vous pouvez voir que le nombre de budget à changer puisque dans notre exemple celui-ci est passé à 500010 qui est la somme de (10 x la quantité) + 500000. De plus, lorsque vous cliquez sur une vente vous pouvez voir les détails de cette vente (cercle bleu).

Vous savez maintenant ajouter une vente, vous pouvez tester et créer vos dépenses du moment qu'elles restent raisonnables et au dessus de 0 sinon une erreur se produira.

Vous savez maintenant créer des ventes et des dépenses pour votre festival. Vous allez pouvoir gérer votre argent de la meilleur façon et pouvoir gérer vos bénéfices. Grâce à cela, vous ne pourrez plus vous faire arnaquer !

V-Gestion des erreurs

Cette partie va grandement vous aider puisque si vous avez une erreur qui apparaît, c'est ici que vous trouver les réponses.

1. Les erreurs de saisie

Ces erreurs sont mineures voire inexistantes si vous suivez la documentation à la lettre. Mais si jamais cela arrive, nous vous conseillons de regarder le schéma ci-dessous qui est une matrice test concernant les saisies et leur validation.

Matrice de saisie pour la fenêtre festival:

	Zone de saisie vide	valeur saisie numérique	valeur saisie non numérique
Saisie du nom	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
Saisie du budget	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect,seul des entiers ou réels sont acceptés" Zone de saisie actualisée Zone de résultat = Vide
Saisie du nb place	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user	Message d'erreur = "Caractère incorrect,seul des entiers sont acceptés" Zone de saisie actualisée Zone de résultat = Vide
Saisie de la Date	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user	Message d'erreur = "Caractère incorrect,seul des entiers sont acceptés" Zone de saisie actualisée Zone de résultat = Vide

Saisie des organisateur	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
saisie du nb artistes	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user	Message d'erreur = "Caractère incorrect,seul des entiers sont acceptés" Zone de saisie actualisée Zone de résultat = Vide
Clic sur enregistrer	Message d'erreur = "une des zones de saisie est vide" Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "une des zones de saisie est fausse(erreur de type détecté)" Zone de saisie actualisée Zone de résultat de l'erreur = Vide	Pas de message d'erreur Zone de saisie actualisée et sauvegarde des informations

Matrice de saisie pour la fenêtre de création d'un produit :

	Zone de saisie vide	valeur saisie numérique	valeur saisie non numérique
Saisie du libellé	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
Saisie de la référence	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
saisie du prix	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user	Message d'erreur = "Caractère incorrect,valeur num attendue" Zone de saisie actualisée Zone de résultat = Vide

saisie de l'utilité	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
saisie de la marque	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
Clic sur enregistrer	Message d'erreur = "une des zones de saisie est vide" Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "une des zones de saisie est fausse(erreur de type détecté)" Zone de saisie actualisée Zone de résultat de l'erreur = Vide	Pas de message d'erreur Zone de saisie actualisée et sauvegarde des informations
	zone vide avant click	zone vide après click	
choix du type de produit (dépense ou ticket)	Message d'erreur = "Choix obligatoire du type de produit" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée et sauvegarde des informations	

Matrice de saisie pour la fenêtre de création d'un artiste :

	Zone de saisie vide	valeur saisie numérique	valeur saisie non numérique
zone de saisie du nom	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user

zone de saisie du genre musicale	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
zone de saisie du cachet	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user	Message d'erreur = "Caractère incorrect, valeur num attendue" Zone de saisie actualisée Zone de résultat = Vide
	zone vide avant click	zone vide après click	
choix du jour	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Choix obligatoire du type de produit" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
choix de la durée	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Choix obligatoire du type de produit" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
choix de l'heure	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Choix obligatoire du type de produit" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
liste déroulante des scènes	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Choix obligatoire du type de produit" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
Clic sur enregistrer	Message d'erreur = "une des zones de saisie est vide" Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "une des zones de saisie est fausse (erreur de type détecté)" Zone de saisie actualisée	Pas de message d'erreur Zone de saisie actualisée et sauvegarde des

		Zone de résultat de l'erreur = Vide	informations
--	--	-------------------------------------	--------------

Matrice de saisie pour la fenêtre de création d'un stand :

	Zone de saisie vide	valeur saisie numérique	valeur saisie non numérique
saisie du nom	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
saisie de la longueur	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user	Message d'erreur = "Caractère incorrect,valeur num attendue" Zone de saisie actualisée Zone de résultat = Vide
saisie de la largeur	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user	Message d'erreur = "Caractère incorrect,valeur num attendue" Zone de saisie actualisée Zone de résultat = Vide
saisie du champs spécifique	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "Caractère incorrect, les entiers ou réel ne sont pas acceptés" Zone de saisie actualisée Zone de résultat = Vide	Pas de message d'erreur Zone de saisie actualisée Zone de résultat = saisie de l'user
Clic sur enregistrer	Message d'erreur = "une des zones de saisie est vide" Zone de saisie actualisée Zone de résultat = Vide	Message d'erreur = "une des zones de saisie est fausse(erreur de type détecté)" Zone de saisie actualisée Zone de résultat de	Pas de message d'erreur Zone de saisie actualisée et sauvegarde des informations

		l'erreur = Vide	
--	--	-----------------	--

Figure 17 : *Matrice test*

Ensuite une erreur que vous pouvez trouver dès le début lors d'un clic sur un module autre que « paramètre festival » :

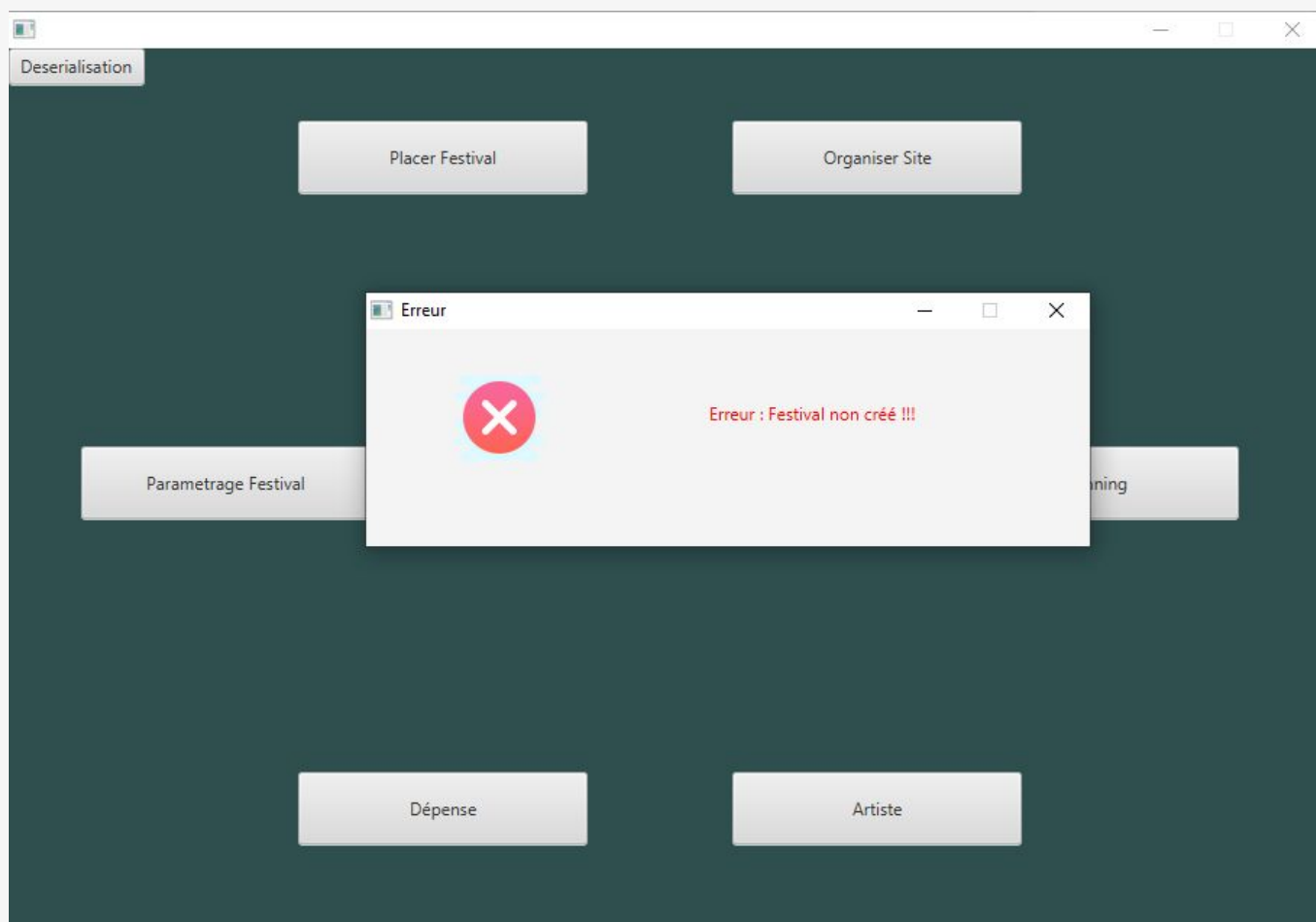


Figure 18 : Fenêtre *d'erreur*

Si cette fenêtre apparaît, ne vous en faites pas puisque il indique juste que vous ne pouvez utiliser les autres modules seulement si le paramétrage du festival a été fait. Pour supprimer cette erreur, il vous suffit de cliquer sur la croix en haut à gauche de la fenêtre d'erreur et de cliquer sur le paramétrage du festival.

Également si une fenêtre ne s'affiche pas correctement, fermer celle-ci puis rouvrez là. Si le problème persiste fermer l'application et ouvrez-la à nouveau.

Si des problèmes techniques importants apparaissent (mauvaise affichage pour placer le festival ou les éléments, le système pour générer la ville dysfonctionne...), contactez-nous pour que l'on puisse régler ces problèmes :

Contact des développeurs :

- Breton Paul : paul.breton@etudiant.univ-rennes1.fr
- Plestan Corentin : corentin.plestan@etudiant.univ-rennes1.fr
- Fouillé Justine : justine.fouille@etudiant.univ-rennes1.fr
- Tairi Glory-Sara : glory-sara.tairi@etudiant.univ-rennes1.fr
- Mathurin Melvin : melvin.mathurin@etudiant.univ-rennes1.fr