

SPECIFICATIONS TECHNIQUES

Groupe 21 :

- KOTANMI Amédé Florian
- LEGBASSI-GODONOU Godfried
- SEWADE Fadel

I. Spécifications techniques du projet

I. 1 - Contexte et définition du projet

Pour diverses raisons, un lieu de rassemblement demeure indispensable pour une bonne tenue des activités humaines. Selon l'évènement que vous souhaitez tenir, la prise d'une salle se présente à vous comme incontournable. Face à la croissance des demandes de salles, on remarque nombre de désorganisations. Pour cela, la réservation des salles est devenu rapidement une solution pour pallier à ces divers problèmes.

Le gestion des salles permet aux clients de retrouver et de réserver diverses salles pour leur évènements. Un logiciel de réservation des salles moderne élimine :

- ➔ la conjecture dans la planification ;
- ➔ l'annulation et le rééchelonnement des salles.

Cela simplifie le processus de réservation. En plus, un logiciel de réservation des salles habilite les responsables des locaux avec des connaissances approfondies sur le statut d'occupation et l'utilisation des salles. Ainsi, les entreprises (le gestionnaire) développe(nt) une gestion globale d'usage d'espace, en prenant des décisions éclairées sur la répartition des locaux, la création d'espaces hot-desking etc.

I.2 - Objectif du projet

Ce projet vise principalement à remédier à deux problèmes. Ci-dessus énoncés il s'agit de :

- ➔ L'élimination de la conjecture dans la planification ;
- ➔ l'annulation et le rééchelonnement des salles ;

I.3 - Périmètre du projet

Ce projet ne dispose pas de restriction, de limites à ne pas dépasser. Un des conseils prodigés par le professeur est l'innovation. Cela paraît pour nous comme étant une autorisation à déborder d'imagination.

I.4 - Description fonctionnelle des besoins

En termes de fonctionnalité, le logiciel à concevoir doit pouvoir :

- ✕ inscrire des clients ;
- ✕ Permettre la connexion des clients à leurs différents comptes ;
- ✕ permettre l'administration du logiciel par un seul administrateur (gérant) ;
- ✕ réserver les salles disponibles ou non ;
- ✕ lister les différentes salles ainsi que leur statut (réservé ou non), de même que les différents prix associés;

Pour atteindre cet objectif, l'architecture générale utilisée est l'architecture orienté objet. Les composants logiciels qui seront mis à contribution sont :

- ◆ Comme éditeur de code, nous utiliserons CodeBlocks 20.03 ;
- ◆ Le langage de programmation utilisé est le C++, accompagné de la bibliothèque graphique wxWidgets;
- ◆ Dans la mesure du possible, nous tenterons l'utilisation d'une base de données afin d'y stocker les username ainsi que les passwords des utilisateurs ;
- ◆ Les données personnelles : La gestion et la protection des données étant cruciale de nos jours, la sécurisation de telles données sera prise en compte.

● Les acteurs du logiciel :

Le Gérant:

- ✕ A accès à l'application via un password propre à lui;
- ✕ peut consulter la liste des salles réservées ou non;
- ✕ a également accès aux informations de tous les clients;
- ✕ peut retirer les utilisateurs du logiciel en cas de besoin;
- ✕ peut annuler n'importe quelle réservation ;

Le Client :

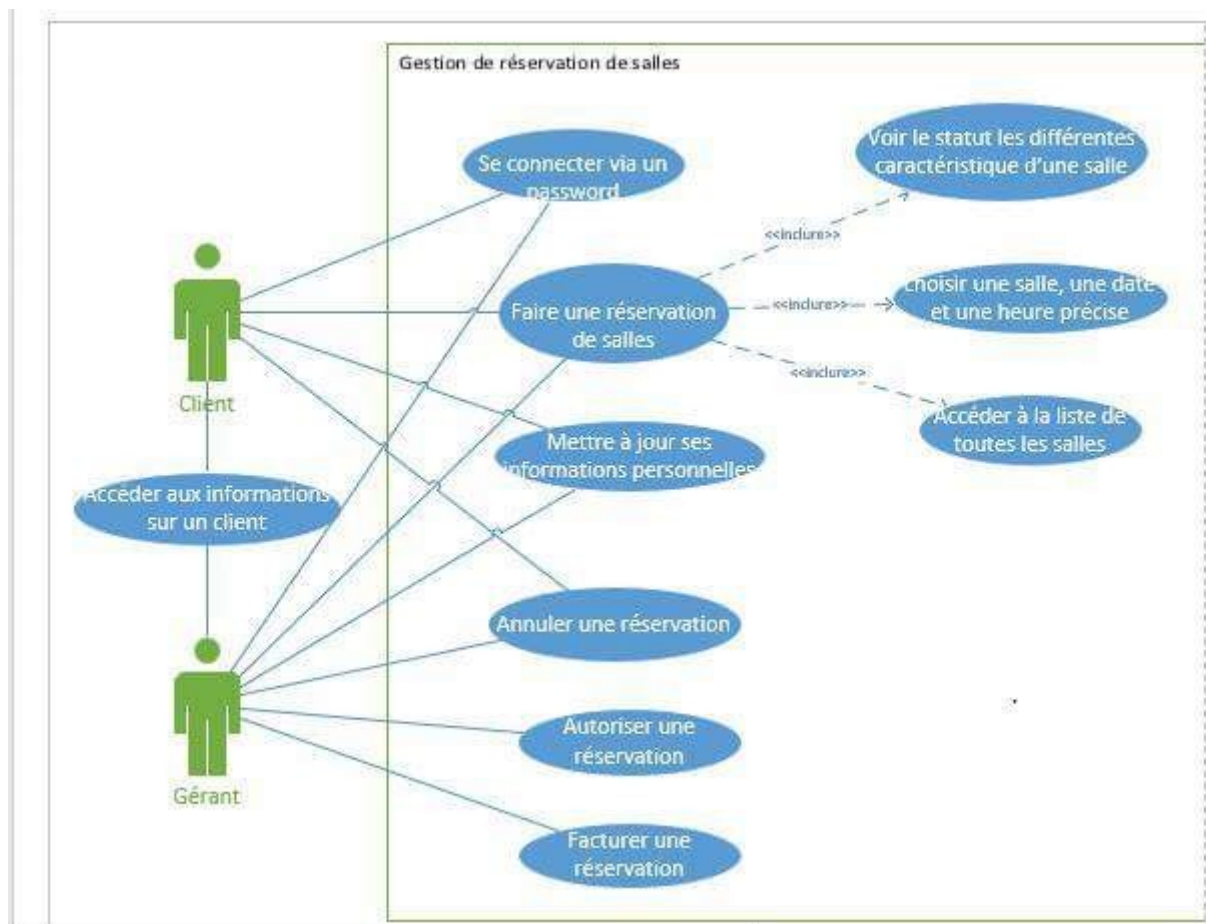
- ✕ s'identifie dans le logiciel à l'aide d'un username et d'un password;
- ✕ a la possibilité de réserver des salles à n'importe quelle heure;
- ✕ a accès à des salles réservées ou non;
- ✕ a la possibilité d'annuler sa réservation

I.6 - Délais de réalisation

Aucun délai de réalisation n'a été précisé pour cette réalisation. Elle se déroule suivant différentes phases définies par le professeur.

I.7 - Diagrammes de cas d'utilisation

Comme diagramme de cas d'utilisation, nous présenterons ce qui suit comme étant une utilisation globale de notre logiciel par les acteurs concernés (le gérant qui fait office d'administrateur et les clients).



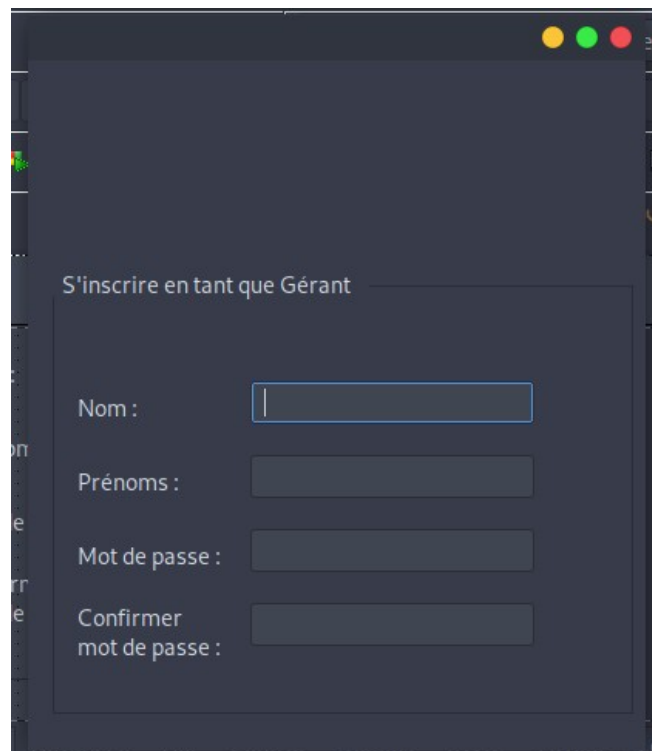
II. Schémas d'Ecran

Ci dessous les différents schémas d'écran disponibles.

- **Ecran d'Inscription :**



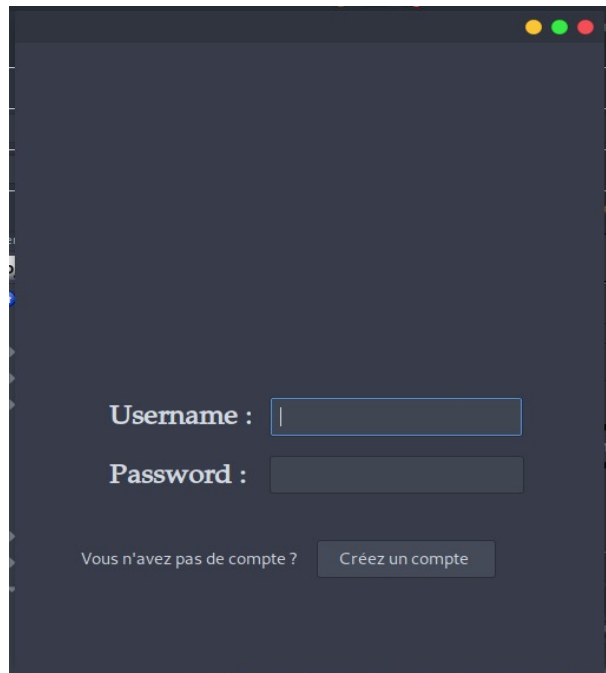
- **Ecran d'inscription du Gérant**



Ceci représente l'écran sur lequel le gérant en d'autre termes, l'administrateur du logiciel s'inscrit. Il ne peut qu'y avoir un seul gérant. C'est la première fenêtre

qui apparaît dans le logiciel. Notons à ce niveau qu'un identifiant sous la forme « #nom_gerant » est généré par le logiciel afin de le différencier des autres utilisateurs lambda.

- **Ecran de connexion à son compte**

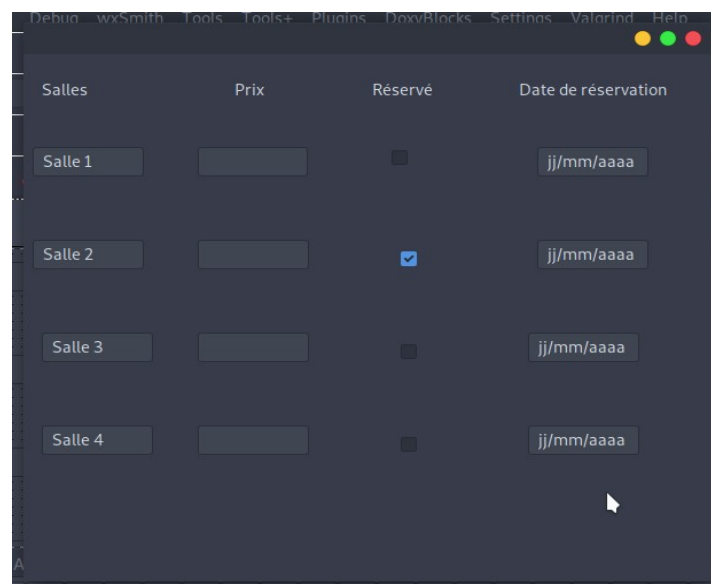


Username :

Password :

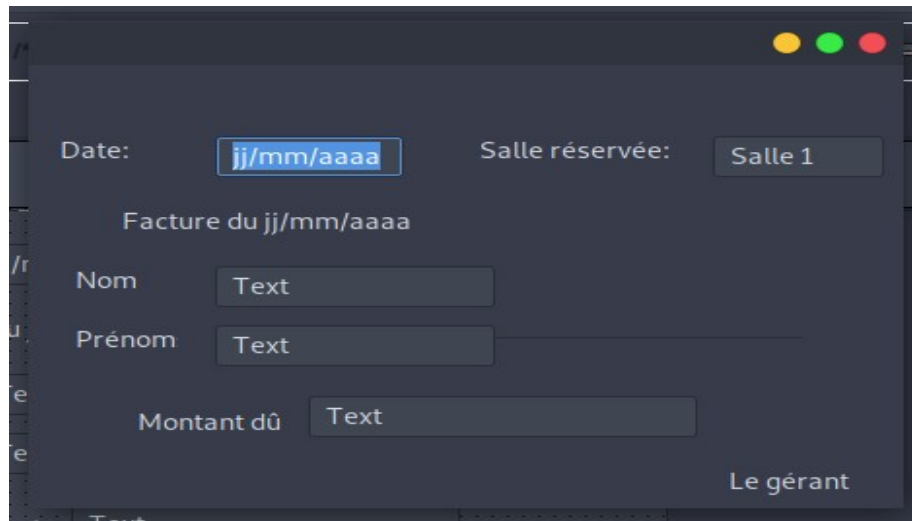
[Vous n'avez pas de compte ?](#)

- **Liste des salles s'affichant sur l'interface du client**



Salles	Prix	Réservé	Date de réservation
Salle 1	<input type="text"/>	<input type="checkbox"/>	jj/mm/aaaa
Salle 2	<input type="text"/>	<input checked="" type="checkbox"/>	jj/mm/aaaa
Salle 3	<input type="text"/>	<input type="checkbox"/>	jj/mm/aaaa
Salle 4	<input type="text"/>	<input type="checkbox"/>	jj/mm/aaaa

- **Ecran de facturation :**

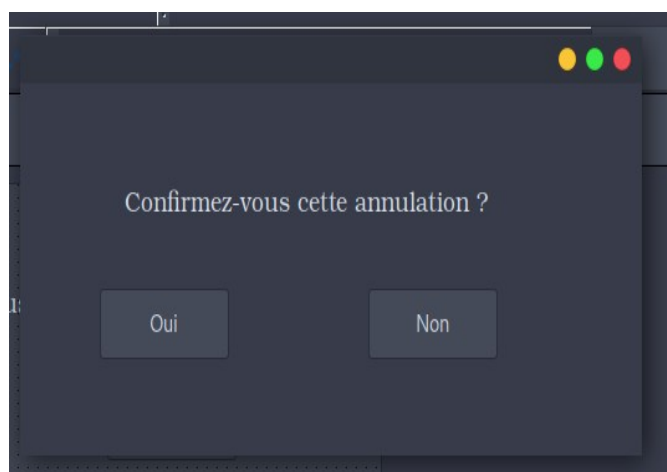
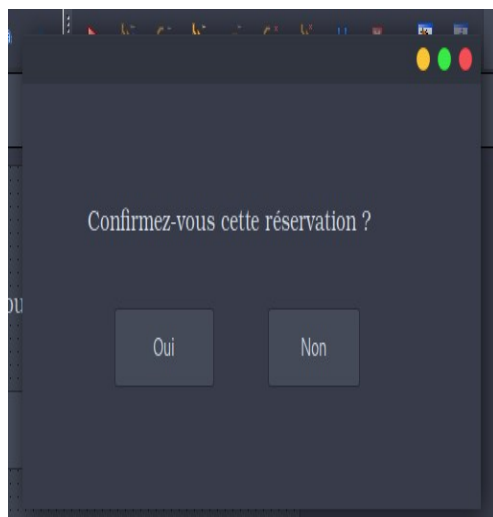


A dark-themed window titled "Facture" (Invoice) with standard macOS window controls (yellow, green, red buttons) in the top right corner. The form contains the following elements:

- Date:** A text input field containing "jj/mm/aaaa".
- Salle réservée:** A text input field containing "Salle 1".
- Facture du jj/mm/aaaa**: A label indicating the invoice date.
- Nom**: A text input field with the placeholder "Text".
- Prénom**: A text input field with the placeholder "Text".
- Montant dû**: A text input field with the placeholder "Text".
- Le gérant**: A label at the bottom right of the form area.

- **PopUp de validation**

Il s'agit des petites boites de dialogues qui s'affichent pour confirmer le choix d'un client



III. Diagrammes de classes

