

Réprésentation de TP INF 4198 : Génie Logiciel

THEME : APPLICATION DE LOCATION DE SALLE D'ÉVÈNEMENT

Introduction générale

Dans le monde dynamique et organisé d'aujourd'hui, la gestion et l'utilisation efficaces des espaces sont cruciales pour les individus et les organisations.

Qu'il s'agisse de planifier des réunions, de programmer des événements ou de coordonner l'utilisation de l'espace de travail, une application de réservation de salles bien structurée peut rationaliser le processus et garantir une allocation optimale des ressources.

La question que se pose des personnes est la suivante : **comment trouver rapidement une salle appropriée à mon événement ?**

Analyse du problème

- Avant l'ère des applications, lorsque une personne voulait une salle pour organiser ses événements, il devait soit se renseigner à son entourage et connaissance pour savoir les salles appropriées pour lui et leur zone géographique.
- Où il devait marcher zone par zone à la recherche de salles et appeler à chaque fois le propriétaire pour avoir les informations sur la salle, voir visité afin de s'assurer qu'elle correspondait à ses attentes.
- Les réservations et la planification des salles étaient gérées manuellement, souvent au moyen de calendriers physiques, de feuilles de calcul ou de communication verbale.
- Le développement de plateformes telles que **Event Place** et **IZIPARTY**, qui permettent aux utilisateurs de rechercher la salle appropriée à leur besoin et de faire une réservation en ligne existent mais ne prennent pas en compte l'analyse multi-critère.

Problématique :

Comment concevoir et développer une application web conviviale et performante qui permette à la fois aux organisateurs d'événements de réserver facilement des salles adaptées à leurs besoins, et aux propriétaires de salles de gérer efficacement leur offre et leurs réservations, dans un contexte de marché concurrentiel ?

Objectif Spécifique:

- Permettre une recherche et une exploration faciles du catalogue de salles disponibles à la location.
- Offrir un processus de vérification de disponibilité et de réservation en ligne simplifié.
- Développer une interface utilisateur intuitive pour la gestion des réservations (consultation, modification, annulation).
- Côté propriétaires de salles :
- Fournir un outil de gestion complète du catalogue de salles (création, modification, suppression).
- Développer un système de suivi et de validation des réservations pour les propriétaires.
- Améliorer la visibilité et la location des salles proposées sur la plateforme.

Questions de recherche

- Quelles sont les principales attentes des organisateurs d'événements en matière de réservation de salles ?
- Comment faciliter le processus de création et de mise à jour du catalogue de salles ?
- Quelles fonctionnalités de gestion des réservations sont les plus pertinentes pour les utilisateurs ?
- Quel processus de réservation en ligne permet de maximiser la fluidité et la simplicité d'utilisation?

Champ d'application :

- L'application sera initialement déployée dans la ville de Yaoundé, qui constitue un marché important et concurrentiel pour la location de salles d'événements.
- À terme, l'objectif sera d'étendre le service à l'ensemble du territoire Camerounais, puis potentiellement à d'autres pays d'Afrique.
- Organisateurs d'événements : entreprises, associations, particuliers organisant des événements professionnels ou privés (conférences, séminaires, soirées, mariages, etc.).
- Propriétaires de salles : hôtels, centres de conférences, salles polyvalentes, lieux atypiques, etc.

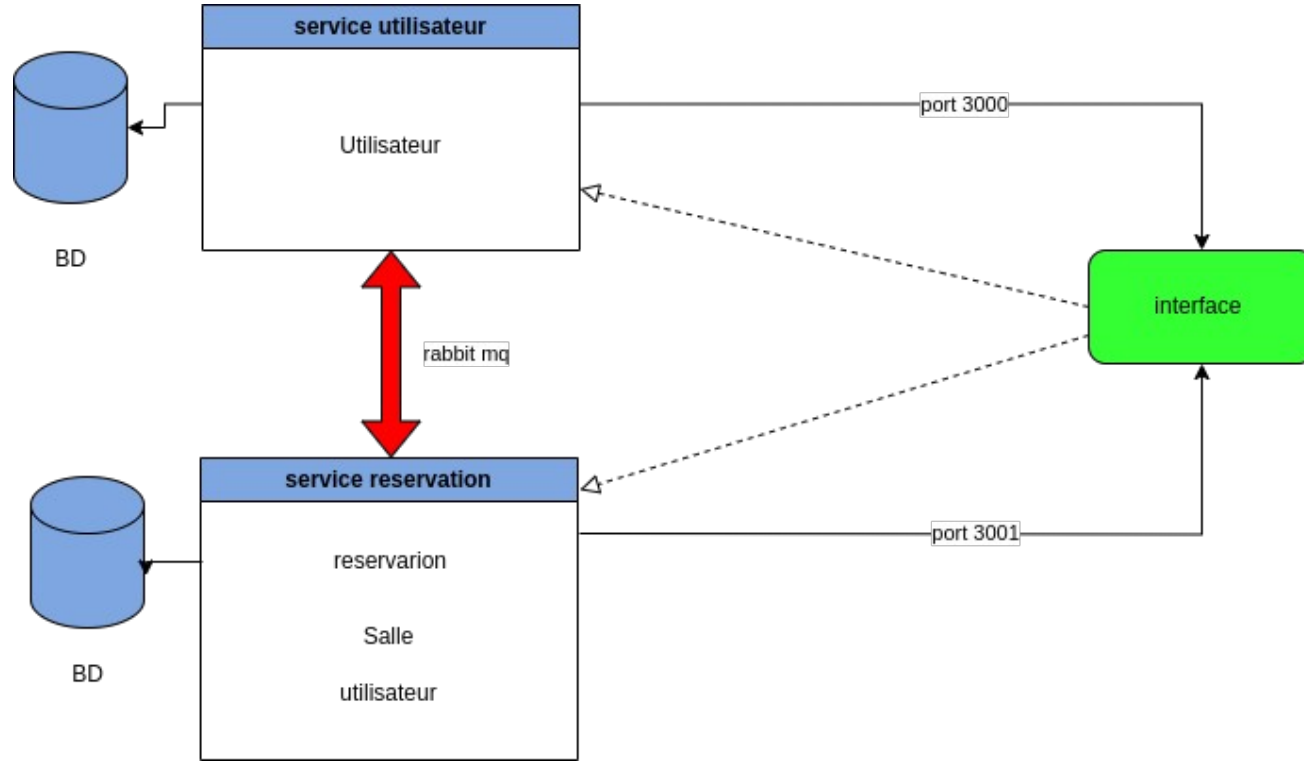
Exigences fonctionnelles.

- Permettre aux organisateurs d'événements de rechercher des salles à louer à partir de critères avancés (localisation, capacité, équipements, styles, etc.)
- Proposer un processus de réservation,
- Offrir la possibilité de gérer les réservations (modification, annulation, etc.),
- Permettre aux propriétaires de salles de créer et gérer leurs fiches de présentation (description, photos, équipements, tarifs, etc.),
- Permettre aux organisateurs d'événements de recevoir une recommandation de salles en fonction de leur critères de préférence.

Exigences non-fonctionnelles

- Réactivité: le temps de réponse des requêtes faites au système sera d'au moins 50 secondes.
- Accessibilité: Notre service sera accessible via une application web.
- Disponibilité : Notre service garantie une disponibilité de 98,9%.
- Fiabilité : les opérations du système sont hautement fiables.
- Robustesse : les opérations de recherche sont robustes aux pannes du système (tolérance aux pannes).

Architecture Microservice de l'application HallHunt



Modèle mathématique

Ensembles :

S : Ensemble des salles : représente l'ensemble des salles i disponibles pour la location.

$$S=(1,...,i)$$

C : Ensemble des clients : représente l'ensemble des clients j souhaitant louer une salle.

$$C=(1,...,j)$$

T: Ensemble des créneaux horaires : représente l'ensemble des créneaux horaires t disponibles pour la réservation.

$$T=(1,...,t)$$

Paramètres:

Capacité(i) : capacité maximale de la salle.

TempsMinimum(j): de temps minimum de location de salle.

Variables de Décision :

Variable binaire " $x(i, j, t)$ " : représente la variable de décision indiquant si la salle " i " est attribuée au client " j " au créneau horaire " t ".

La valeur de " $x(i, j, t)$ " est 1 si la salle est attribuée, et 0 sinon.

Modèle mathématique

Contraintes :

Contrainte de disponibilité : chaque salle ne peut être attribuée qu'à un seul client au même créneau horaire.

Contrainte de capacité : la capacité de la salle doit être suffisante pour accueillir le nombre de participants du client.

Contrainte de préférence : certains clients peuvent avoir des préférences spécifiques pour certaines salles ou créneaux horaires.

Contrainte de temps minimum de location : certains clients peuvent exiger un temps minimum de location pour une salle donnée.

Objectif :

Maximiser l'utilisation des salles : l'objectif est de maximiser l'utilisation des salles en attribuant les clients aux salles et créneaux horaires appropriés, tout en respectant les contraintes.

Modélisation mathématique :

Soit "S" l'ensemble des salles, "C" l'ensemble des clients, et "T" l'ensemble des créneaux horaires.

La fonction objectif peut être exprimée comme :

Maximiser $\sum(i, j, t) [x(i, j, t)]$

où (i, j, t) appartient à (S, C, T)

Sous réserve des contraintes suivantes :

Pour chaque salle "i" et créneau horaire "t", la capacité maximale de la salle doit être respectée :

$\sum(j) [x(i, j, t)] \geq \text{Capacité}(i)$

Modèle mathématique

Pour chaque client "j" et créneau horaire "t", la salle attribuée doit respecter les préférences du client :
 $x(i, j, t) = 0$ si la salle "i" n'est pas une préférence du client "j" ou n'est pas disponible au créneau horaire "t".

Pour chaque client "j", la contrainte de disponibilité s'applique :
 $\sum_t [x(i, j, t)] = 1$

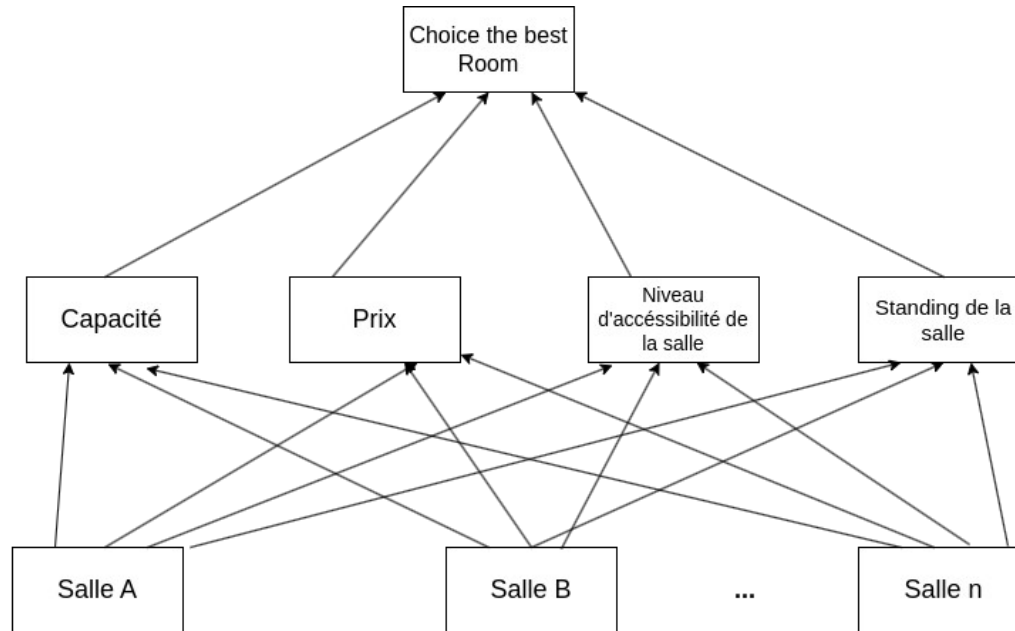
Pour chaque client "j", la contrainte de temps minimum de location peut être exprimée comme :
 $\sum_t [x(i, j, t)] \geq \text{TempsMinimum}(j)$ pour au moins une salle "i".
Les variables " $x(i, j, t)$ " sont binaires (0 ou 1) et représentent l'attribution des clients aux salles et créneaux horaires.

L'approche méthodologique

La méthode AHP (Analytic Hierarchy Process) est une approche multicritère qui peut être utilisée pour évaluer et sélectionner les meilleures salles de location d'événements en fonction de différents critères. Voici les principales étapes de la méthodologie AHP dans ce contexte :

- 1) Structurer ces critères de manière hiérarchique (objectif principal, critères principaux, sous-critères).
- 2) Identifier les critères les plus pertinents pour l'évaluation des salles, par exemple : capacité, équipements, accessibilité, prix, qualité des services,
- 3) Attribuer des poids relatifs à chaque critère en fonction de leur importance respective pour l'organisateur d'événement.
- 4) Calcul des scores globaux (poids de critères)
- 5) Vérifier la consistance des valeurs
- 6) Classement et sélection

Modèle hiérarchique du système



HIERARCHICAL MODEL FOR CHOICE THE BEST ROOM