2025-02-09

Jeremy Guerin

CEGEP DE CHICOUTIMI

Conception détaillée

DÉVELOPPEMENT D'APPLICATIONS ET ESP

Table des matières

[Cahier de conception détaillée pour la Gestion du Système de Tickets de Stationnement 2](#_Toc189985266)

[Diagramme de base de données 2](#_Toc189985267)

[Diagramme de cas d’utilisation 3](#_Toc189985268)

[Diagrames de séquence de la borne de paiement 0](#_Toc189985269)

[Principaux cas d'utilisation du système : 0](#_Toc189985270)

[1. Génération d'un ticket de stationnement (Borne d’Entrée) 0](#_Toc189985271)

[2. Paiement du ticket (Borne de Paiement) 0](#_Toc189985272)

[3. Validation du ticket et sortie du stationnement (Borne de Sortie) 1](#_Toc189985273)

[4. Gestion des tickets et des utilisateurs (Logiciel d’Administration) 2](#_Toc189985274)

[5. Gestion des abonnements et des codes de réduction (Logiciel d’Administration) 2](#_Toc189985275)

[6. Génération et confirmation des tickets (Server) 3](#_Toc189985276)

[7. Débogage et vérification des calculs (Logiciel d’Administration) 3](#_Toc189985277)

[Maquettes du logiciel d’administration 4](#_Toc189985278)

[Maquettes de la borne de paiement 6](#_Toc189985279)

[Composants 6](#_Toc189985280)

[Composants matériels : 6](#_Toc189985281)

[Composants logiciels : 7](#_Toc189985282)

[Représentation graphique des composants du système 8](#_Toc189985283)

[Plan de test 9](#_Toc189985284)

[1. Tests de la Borne d’Entrée 9](#_Toc189985285)

[2. Tests de la Borne de Paiement 9](#_Toc189985286)

[3. Tests de la Borne de Sortie 10](#_Toc189985287)

[4. Tests du Logiciel d’Administration 11](#_Toc189985288)

[5. Tests du Serveur 11](#_Toc189985289)

[6. Tests de Sécurité 12](#_Toc189985290)

[7. Tests de Performance 12](#_Toc189985291)

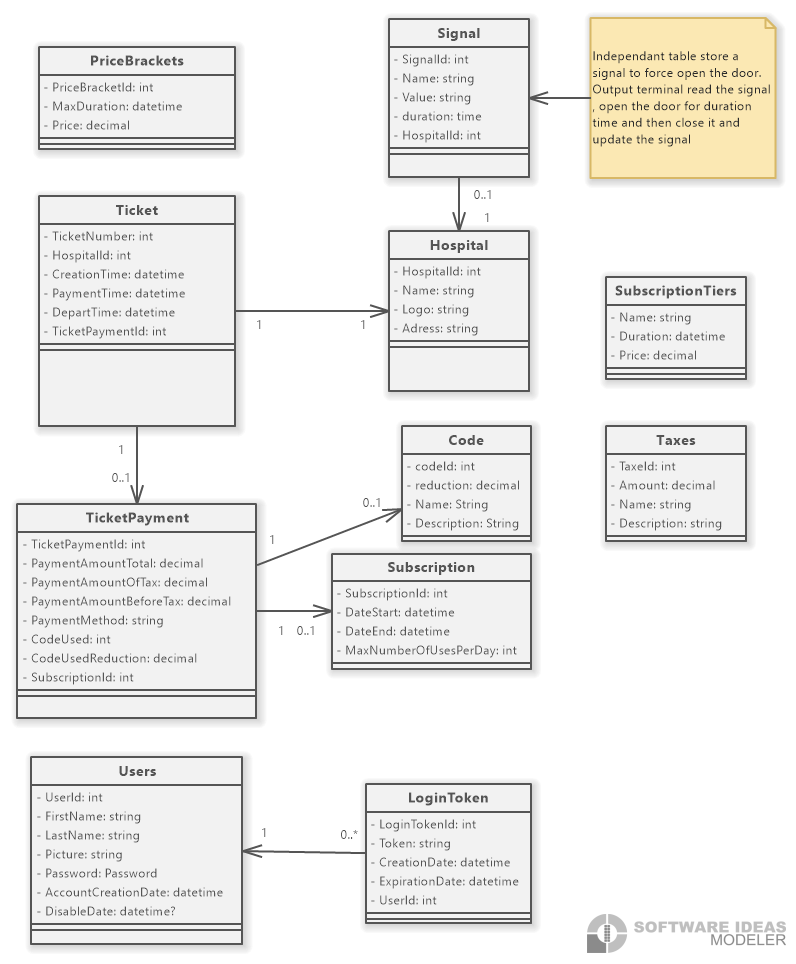
[8. Tests d’Interface Utilisateur 13](#_Toc189985292)

[9. Tests de Robustesse 13](#_Toc189985293)

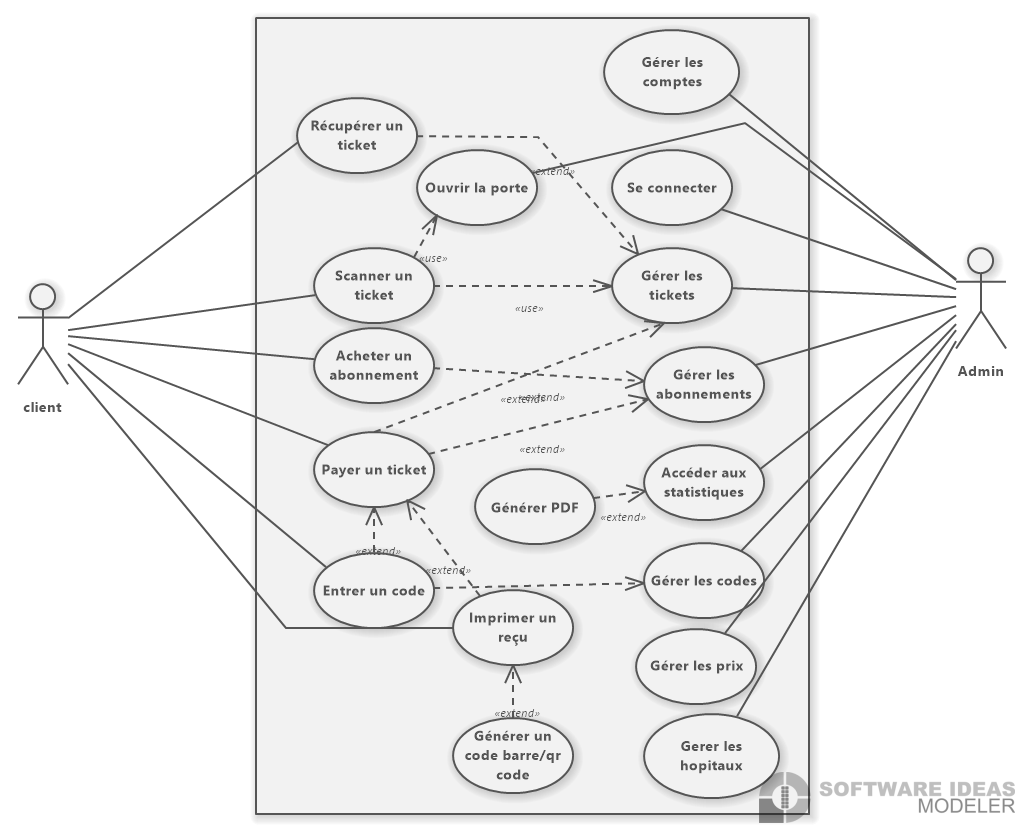
[Autres requis 14](#_Toc189985294)

# Cahier de conception détaillée pour la Gestion du Système de Tickets de Stationnement

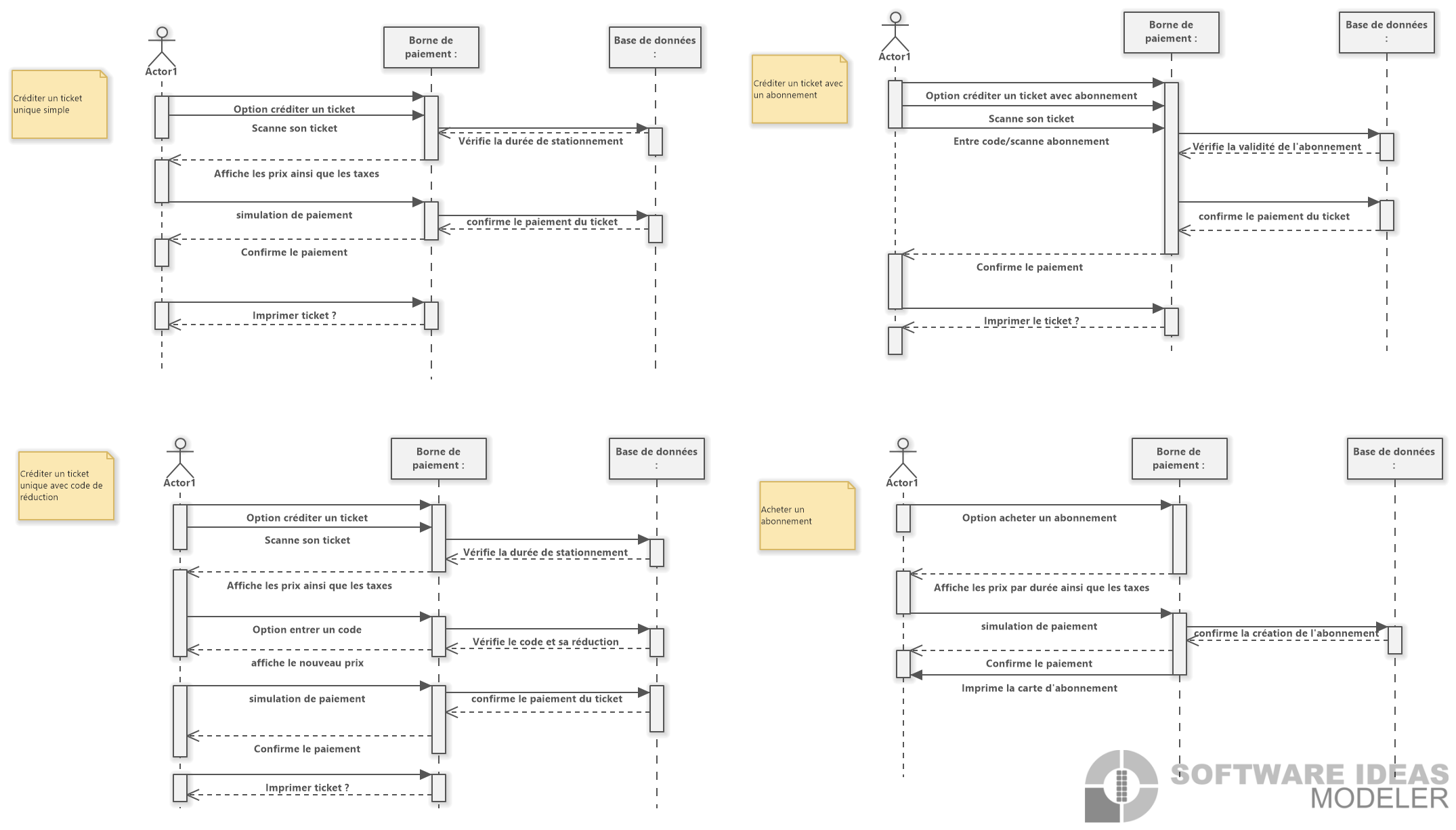
## Diagramme de base de données



## Diagramme de cas d’utilisation



## Diagrames de séquence de la borne de paiement



## Principaux cas d'utilisation du système :

### 1. Génération d'un ticket de stationnement (Borne d’Entrée)

- **Acteur principal** : Utilisateur du stationnement (conducteur).

- **Description** :

1. L'utilisateur arrive à la borne d’entrée du stationnement et appuie sur un bouton pour générer un ticket.
2. La borne imprime un ticket contenant les informations suivantes :
   1. Identité de l’hôpital (nom, logo).
   2. Heure d’arrivée.
   3. Identifiant unique du ticket.
   4. Code-barres ou QR code associé.
3. La barrière d’entrée s’ouvre automatiquement après la génération du ticket.

- **Précondition** :

- La borne d’entrée est opérationnelle et connectée via LTE.

- **Postconditions** :

- Le ticket est généré et enregistré dans la base de données.

- La barrière est ouverte pour permettre l’accès au stationnement.

### 2. Paiement du ticket (Borne de Paiement)

**- Acteur principal** : Utilisateur du stationnement (conducteur).

- **Description** :

1. L'utilisateur se rend à la borne de paiement.
2. Il choisit créditer un ticket unique
3. Il scanne le code-barres ou QR code de son ticket.
4. La borne calcule le montant à payer en fonction de la durée du séjour.
5. L'utilisateur simule un paiement par carte de crédit.
6. Un reçu est imprimé (optionnel) avec les détails suivants :
   1. Identité de l’hôpital.
   2. Heure d’arrivée, heure de sortie, durée du séjour.
   3. Montant payé et taxes applicables.

- **Cas d'utilisation alternatif :**

- Si l'utilisateur entre un code de réduction à l’étape 4 le nouveau montant sera affiché

- L’utilisateur choisit crédit un ticket avec abonnement, scanne sa carte/entre son code d’abonnement, le paiement est automatiquement validé sans frais (dans les limites autorisées).

- **Préconditions** :

- Le ticket a été généré à l’entrée et n’a pas encore été payé.

- La borne de paiement est connectée au réseau local.

- **Postconditions** :

- Le ticket est marqué comme payé dans la base de données.

- Un reçu est imprimé (si l'utilisateur le souhaite).

### 3. Validation du ticket et sortie du stationnement (Borne de Sortie)

- **Acteur** **principal** : Utilisateur du stationnement (conducteur).

- **Description** :

- L'utilisateur se présente à la borne de sortie avec son ticket.

- Il scanne le code-barres ou QR code du ticket.

- La borne vérifie si le ticket a été payé.

- Si le ticket est valide, la barrière de sortie s’ouvre automatiquement.

- **Cas d'utilisation alternatif** :

- Si le ticket n’a pas été payé, la borne affiche un message d’erreur et refuse l’ouverture de la barrière.

- **Préconditions** :

- Le ticket a été généré à l’entrée et payé (ou validé par un abonnement).

- Le ticket est toujours valide (une heure après le paiement)

- La borne de sortie est connectée via LTE.

- **Postconditions** :

- Le ticket est marqué comme utilisé dans la base de données.

- La barrière de sortie est ouverte.

### 4. Gestion des tickets et des utilisateurs (Logiciel d’Administration)

- **Acteur principal** : Administrateur du système.

- **Description** :

1. L'administrateur se connecte au logiciel d’administration via une interface sécurisée.
2. Il accède au tableau de bord pour visualiser :
   1. Un graphique en pointe de tarte des tickets actifs non payés.
   2. Un graphique de tendance des revenus sur les sept derniers jours.
3. Il utilise la console de gestion pour :
   1. Ajouter, modifier ou supprimer des utilisateurs.
   2. Modifier les options de tarification (tarif horaire, demi-journée, journée complète).
   3. Ajuster les taux de taxes (fédérale et provinciale).
   4. Gérer la liste de tous les tickets
4. Il consulte et génère des rapports financiers par période, exportables au format PDF.

- **Cas d'utilisation alternatif** :

- L'administrateur peut désactiver manuellement un ticket perdu (par exemple, le soir lorsque le parking est vide).

- Il peut ouvrir manuellement les barrières d’entrée et de sortie en cas de besoin.

- **Préconditions** :

- L'administrateur dispose des droits d’accès nécessaires.

- Le logiciel d’administration est connecté au réseau local.

- **Postconditions** :

- Les modifications apportées par l'administrateur sont enregistrées dans la base de données.

- Les rapports sont générés et disponibles pour consultation.

### 5. Gestion des abonnements et des codes de réduction (Logiciel d’Administration)

- **Acteur principal** : Administrateur du système.

- **Description** :

1. L'administrateur gère les abonnements mensuels ou annuels pour les usagers réguliers.
2. Il modifie les prix d’abonnements et leurs durées
3. Il crée des codes de réductions et leurs limites (temp, nombre utilisation, utilisable sur abonnements …)
4. Il vérifie l’historique des tickets et désactive les tickets perdus

- **Préconditions** :

- Les fonctionnalités d’abonnement et de codes de réduction sont activées dans le système.

- **Postconditions** :

- Les abonnements et codes de réduction sont enregistrés dans la base de données et peuvent être utilisés par les bénéficiaires.

### 6. Génération et confirmation des tickets (Server)

- **Acteur principal** : Serveur du système.

- **Description** :

1. Le serveur reçoit les demandes des bornes d’entrée et de sortie pour :
   1. Insérer un nouveau ticket dans la base de données lors de la génération à l’entrée.
   2. Confirmer un ticket comme utilisé lors de la validation à la sortie.
2. Le serveur assure la synchronisation des données entre les bornes et la base de données.

- **Préconditions** :

- Le serveur est opérationnel et connecté au réseau local ou cloud.

- **Postconditions** :

- Les tickets sont correctement enregistrés et mis à jour dans la base de données.

### 7. Débogage et vérification des calculs (Logiciel d’Administration)

- **Acteur principal** : Administrateur du système.

- **Description** :

1. L'administrateur consulte un rapport de débogage pour vérifier les calculs de tarification avec précision.
2. Il peut identifier et corriger les erreurs éventuelles dans les calculs de durée et de tarifs.

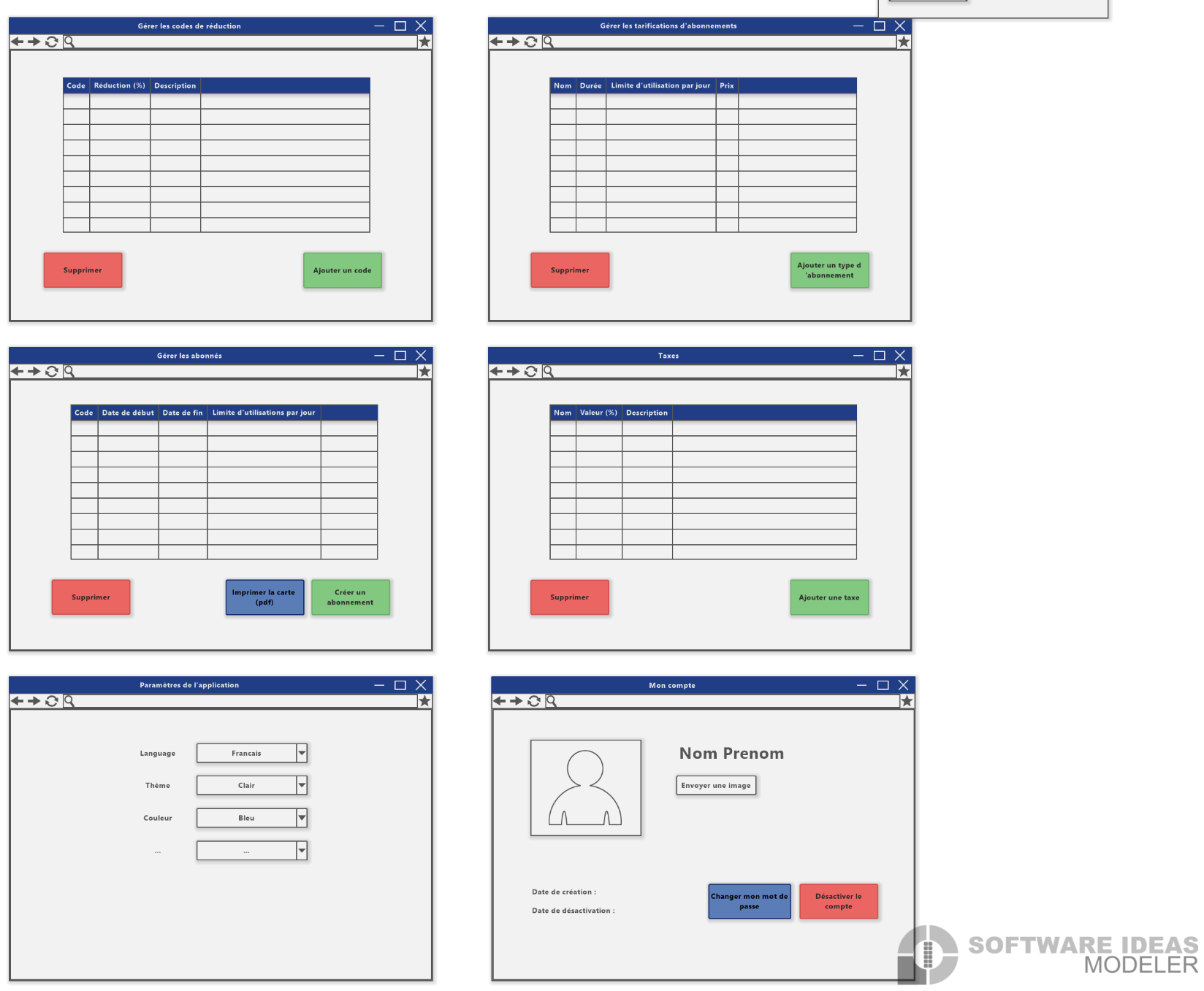
- **Préconditions** :

- Le système a enregistré des données de tickets et de paiements.

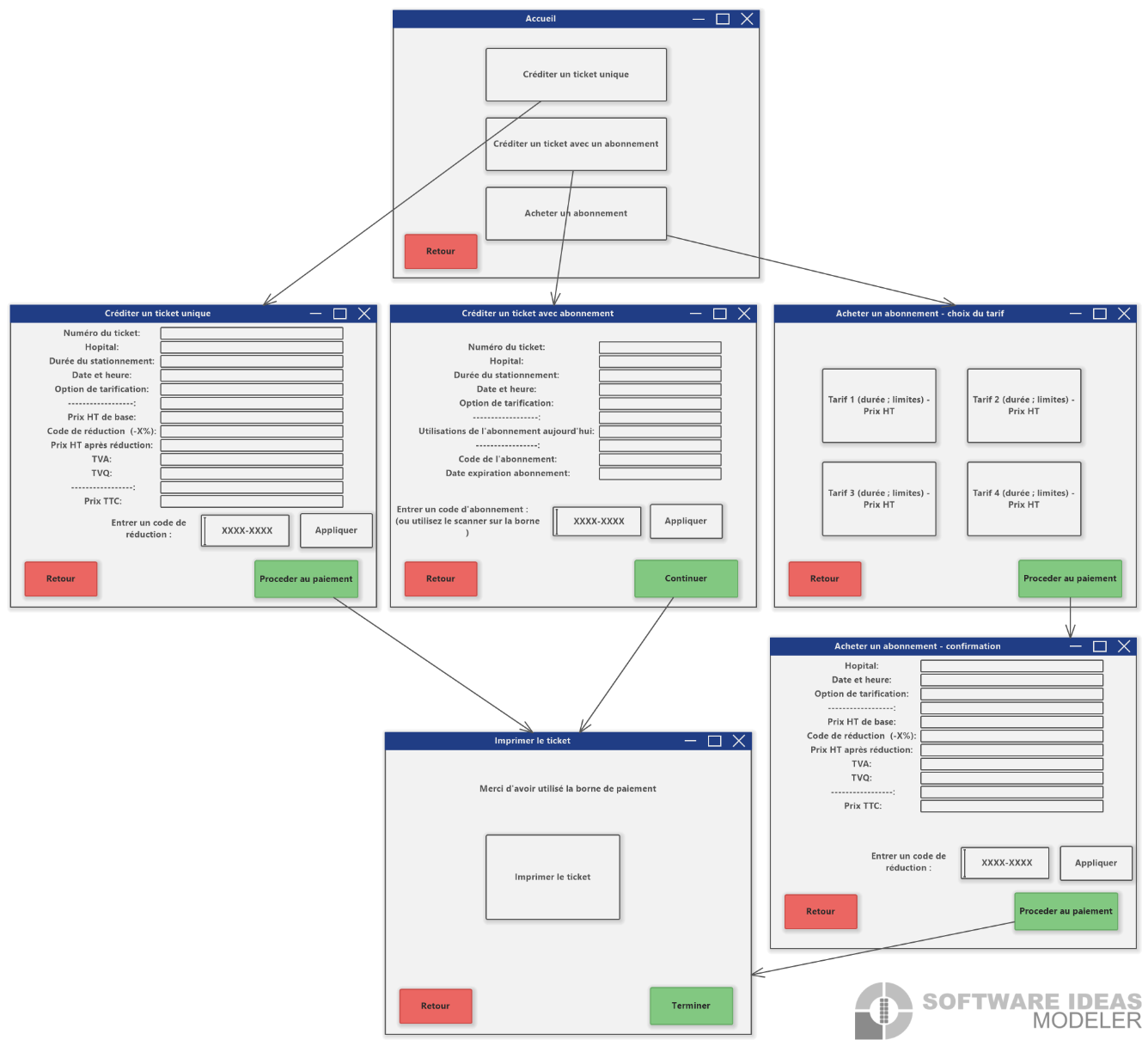
- **Postconditions** :

- Les erreurs de calcul sont identifiées et corrigées.

## A screenshot of a computer AI-generated content may be incorrect.Maquettes du logiciel d’administration



## Maquettes de la borne de paiement



## Composants

### Composants matériels :

**Bornes d'entrée :**

* Écran tactile (résolution 640x480).
* Imprimante de tickets.
* Barrière motorisée.
* Module de communication (LTE).

**Bornes de sortie :**

* Écran tactile (résolution 640x480).
* Lecteur de code-barres ou QR code.
* Barrière motorisée.
* Module de communication (LTE).

**Borne de paiement :**

* Écran tactile.
* Lecteur de code-barres ou QR code.
* Terminal de paiement (simulation).
* Imprimante de reçus.
* Module de communication (réseau local).
* Module d’impression des cartes d’abonnement

**Serveur :**

* Serveur physique ou cloud pour héberger la base de données.
* Module de communication (LTE et réseau local).

**Base de données :**

* Base de données MySql gérée par le cegep

**Réseaux :**

* Réseau LTE
* Réseau local

### Composants logiciels :

**Application de la borne d’entrée :**

* Interface utilisateur (Material Design).
* Module de génération des tickets.
* Module d’impression.

**Application de la borne de sortie :**

* Interface utilisateur (Material Design).
* Module de lecture des tickets (qr code ou code barre).

**Logiciel d'administration :**

* Interface utilisateur (Material Design).
* Tableau de bord avec graphiques (revenus, tickets actifs).
* Module de génération de rapports (PDF).

**Base de données** :

* MySQL pour stocker les données des tickets, des utilisateurs et des transactions.

**Serveur** :

* Application Flask en Python pour gérer les requêtes des bornes.
* API pour la communication entre les bornes et la base de données.

### Représentation graphique des composants du système

A screenshot of a computer

Description automatically generated

## Plan de test

**PLAN DE TEST DÉTAILLÉ POUR LE SYSTÈME DE GESTION DE TICKETS DE STATIONNEMENT**

Le plan de test vise à valider le fonctionnement du système de gestion de tickets de stationnement, en vérifiant chaque composant matériel et logiciel. Les tests seront effectués selon les cas d'utilisation décrits dans le cahier de conception ci-dessus. Voici une description détaillée des tests à réaliser :

### 1. Tests de la Borne d’Entrée

**Objectif** : Vérifier que la borne d’entrée génère correctement un ticket et ouvre la barrière.

**Scénarios de test** :

1. **Génération d'un ticket** :

Action : L'utilisateur appuie sur le bouton de génération de ticket.

Résultat attendu : Un ticket est imprimé avec les informations suivantes :

* + Identité de l’hôpital (nom, logo).
  + Heure d’arrivée.
  + Identifiant unique du ticket.
  + Code-barres ou QR code.
  + La barrière d’entrée s’ouvre automatiquement.
  + Le ticket est enregistré dans la base de données.

1. **Connexion LTE** :

Action : Simuler une perte de connexion LTE.

Résultat attendu : La borne affiche un message d’erreur et refuse de générer un ticket.

1. **Imprimante en panne** :

Action : Simuler une panne de l’imprimante.

Résultat attendu : La borne affiche un message d’erreur et refuse de générer un ticket.

### 2. Tests de la Borne de Paiement

**Objectif** : Vérifier que la borne de paiement calcule correctement le montant à payer, gère les codes de réduction, et valide le paiement.

**Scénarios de test** :

1. **Paiement d'un ticket standard** :

Action : L'utilisateur scanne le code-barres ou QR code du ticket.

Résultat attendu :

* + La borne affiche le montant à payer en fonction de la durée du séjour.
  + L'utilisateur simule un paiement par carte de crédit.
  + Un reçu est imprimé (si demandé).
  + Le ticket est marqué comme payé dans la base de données.

1. **Application d'un code de réduction** :

Action : L'utilisateur entre un code de réduction valide.

Résultat attendu :

* + La borne applique la réduction et affiche le nouveau montant.
  + Le paiement est validé avec le montant réduit.

1. **Paiement avec abonnement** :

Action : L'utilisateur scanne sa carte d’abonnement ou entre son code d’abonnement.

Résultat attendu :

* + Le paiement est validé sans frais (dans les limites autorisées).
  + Le ticket est marqué comme payé dans la base de données.

1. **Ticket non payé** :

Action : L'utilisateur tente de payer un ticket déjà payé ou expiré.

Résultat attendu : La borne affiche un message d’erreur et refuse le paiement.

### 3. Tests de la Borne de Sortie

**Objectif** : Vérifier que la borne de sortie valide correctement les tickets et ouvre la barrière.

**Scénarios de test** :

1. **Validation d'un ticket payé** :

Action : L'utilisateur scanne le code-barres ou QR code d’un ticket payé.

Résultat attendu :

* + La borne vérifie que le ticket est payé et valide.
  + La barrière de sortie s’ouvre automatiquement.
  + Le ticket est marqué comme utilisé dans la base de données.

1. **Ticket non payé** :

Action : L'utilisateur scanne un ticket non payé.

Résultat attendu : La borne affiche un message d’erreur et refuse d’ouvrir la barrière.

1. **Ticket expiré** :

Action : L'utilisateur scanne un ticket payé mais expiré (plus d’une heure après le paiement).

Résultat attendu : La borne affiche un message d’erreur et refuse d’ouvrir la barrière.

### 4. Tests du Logiciel d’Administration

**Objectif** : Vérifier que l’administrateur peut gérer les tickets, les utilisateurs, les tarifs, les abonnements, et générer des rapports.

**Scénarios de test** :

1. **Gestion des tickets** :

Action : L'administrateur consulte la liste des tickets actifs et désactive un ticket perdu.

Résultat attendu : Le ticket est marqué comme désactivé dans la base de données.

1. **Modification des tarifs** :

Action : L'administrateur modifie les tarifs

Résultat attendu : Le nouveau tarif est appliqué lors du calcul du montant à payer.

1. **Gestion des abonnements** :

Action : L'administrateur crée un nouvel abonnement.

Résultat attendu : L’abonnement est enregistré dans la base de données, il peut imprimer la carte qui peut être utilisé par les bénéficiaires.

1. **Génération de rapports** :

Action : L'administrateur génère un rapport financier pour les sept derniers jours.

Résultat attendu : Le rapport est généré au format PDF et contient les données correctes.

### 5. Tests du Serveur

**Objectif** : Vérifier que le serveur gère correctement les requêtes des bornes et synchronise les données.

**Scénarios de test** :

1. **Insertion d'un nouveau ticket** :

Action : La borne d’entrée envoie une demande d’insertion d’un nouveau ticket.

Résultat attendu : Le ticket est enregistré dans la base de données.

1. **Validation d'un ticket** :

Action : La borne de sortie envoie une demande de validation d’un ticket.

Résultat attendu : Le ticket est marqué comme utilisé dans la base de données.

1. **Synchronisation des données** :

Action : Simuler une panne de réseau puis rétablir la connexion.

Résultat attendu : Les données sont synchronisées entre les bornes et la base de données.

### 6. Tests de Sécurité

**Objectif** : Vérifier que le système est sécurisé contre les accès non autorisés et les failles de sécurité.

**Scénarios de test** :

1. **Accès non autorisé au logiciel d’administration** :

Action : Tenter d’accéder au logiciel d’administration sans identifiants valides.

Résultat attendu : L’accès est refusé.

1. **Protection des données utilisateurs** :

Action : Tenter d’accéder aux données des utilisateurs via une injection SQL.

Résultat attendu : La base de données rejette la requête malveillante.

### 7. Tests de Performance

**Objectif** : Vérifier que le système peut gérer un grand nombre d’utilisateurs et de transactions simultanément.

**Scénarios de test** :

1. **Charge élevée** :

Action : Simuler 100 utilisateurs générant des tickets simultanément.

Résultat attendu : Le système fonctionne sans ralentissement ni erreur.

1. **Disponibilité 24/7** :

Action : Tester le système en continu pendant 24 heures.

Résultat attendu : Le système reste opérationnel sans interruption.

### 8. Tests d’Interface Utilisateur

**Objectif** : Vérifier que les interfaces sont conviviales et intuitives.

**Scénarios de test** :

1. **Borne d’entrée** :

Action : Tester la navigation sur l’écran tactile.

Résultat attendu : Les boutons sont réactifs et les instructions sont claires.

1. **Logiciel d’administration** :

Action : Tester la génération de rapports et la modification des tarifs.

Résultat attendu : Les fonctionnalités sont accessibles en quelques clics.

### 9. Tests de Robustesse

**Objectif** : Vérifier que le système résiste aux pannes matérielles et logicielles.

**Scénarios de test** :

1. **Panne de l’imprimante** :

Action : Simuler une panne de l’imprimante de la borne d’entrée.

Résultat attendu : La borne affiche un message d’erreur et refuse de générer un ticket.

1. **Panne du serveur** :

Action : Simuler une panne du serveur.

Résultat attendu : Les bornes affichent un message d’erreur et refusent de fonctionner.

## Autres requis

* L'application doit être sécurisée pour protéger les données des utilisateurs et les transactions financières.
* Les bornes doivent être résistantes aux intempéries et aux actes de vandalisme.
* Le système doit être évolutif pour permettre l'ajout de nouvelles fonctionnalités à l'avenir.
* Les bornes doivent être faciles à installer et à entretenir.
* Le logiciel d'administration doit être accessible à distance pour permettre la gestion à distance du système.
* Le système doit être capable de gérer un grand nombre d'utilisateurs et de transactions simultanées.
* L'application doit être conviviale et intuitive pour les utilisateurs finaux.
* Le système doit être fiable et disponible 24/7 pour garantir un service continu aux utilisateurs.
* Les différentes application C# peuvent être développées dans la même solution ou non, afin d'utiliser au mieux les principes de modularité et de réutilisabilité du code en programmation orientée objet.