

Sommaire

Entrée Parking	2
Description diagramme d'activité	2
Description diagramme de séquence	2
Paielement	4
Description diagrammes d'activité	4
Description diagrammes de séquence	5
Paielement borne	5
Paielement bureau	6
Quitter le parking	7
Description diagramme d'activité	7
Description diagramme de séquence	7
Enregistrement de visite de sécurité	9
Description diagramme d'activité	9
Description diagramme de séquence	9
Enregistrement de fautes & Accord de niveau de service surveillance	10
Description diagramme d'activité	10
Description diagramme de séquence	10
Abonnements	11
Description diagramme d'activité	11
Description diagramme de séquence	11
Autres requêtes de gestion	12
Description diagramme d'activité	12
Description diagramme de séquence	12
Diagramme de classes	13
Uses-cases	14
Annexe	15
Diagramme	15
Description textuelles des cas d'utilisations	33

Entrée Parking

Description diagramme d'activité

L'activité de l'entrée dans le parking se présente comme suit :

La voiture roule jusqu'au capteur, puis lorsque le capteur le remarque ("Detecting car"). Si le parking n'est pas rempli, la borne affiche "Appuyer sur le bouton" ("Press the button" is displayed").

Si c'est un abonné, au lieu d'appuyer sur le bouton, l'abonné introduit sa carte dans la fente de la borne de contrôle ("Insertion of subscription card"), la borne vérifie - en moins de 5 secondes - les conditions nécessaires pour que l'abonné puisse se garer dans le parking avec sa carte d'abonnement, qu'il rentre un jour de semaine, qu'il soit abonné au parking en question, que son abonnement ne soit pas expiré et que son abonnement ne soit pas en cours d'utilisation, c'est-à-dire qu'il ne faut pas que le possesseur de la carte soit enregistré comme rentré, mais pas encore sorti ("Subscription verification").

Si toutes les vérifications sont respectées, la borne rend la carte à l'abonné et enregistre l'heure d'entrée de celui-ci. La barrière se lève, l'abonné entre dans le parking et se dirige vers la place qui lui est attribuée, le capteur détecte que la voiture est passée et la barrière s'abaisse. Le nombre de véhicules dans les places allouées est incrémenté de 1 et on compare le nombre de voitures dans le parking avec la capacité du parking, mais puisqu'ils ont chacun leur place attribuée, ce n'est pas la comparaison n'est pas réellement à prendre en compte, surtout en semaine, puisque c'est à ses moments là qu'ils ont des places allouées.

Si c'est un client ordinaire, il appuie sur le bouton ("Press the button"), la borne imprime le ticket en 5 secondes et affiche le message "Prenez le ticket" ("Take the ticket" is displayed), le client prend le ticket et la barrière se lève, le client entre dans le parking, le capteur détecte quand la voiture est passée et la barrière s'abaisse. Le nombre de véhicules dans les places disponibles est incrémenté de 1 et on compare le nombre de voitures dans le parking avec la capacité du parking. Le client est donc officiellement compté dans le parking.

Si le parking est complet, le client ne peut pas entrer, car la borne affiche le message "Complet" ("Complete" is displayed) et ne distribue pas de ticket ("No ticket distribution"). On en conclut que la barrière ne se lève pas. Lorsqu'il y a de la place qui se libère, la borne affiche de nouveau "Appuyez sur le bouton" et on se retrouve au début de l'enchaînement.

Description diagramme de séquence

La séquence de l'entrée dans le parking se présente comme ceci :

Le capteur ("sensor") détecte le client et demande à la borne si l'entrée est possible (askEntry()).

Si c'est un client ordinaire, on vérifie si le parking est rempli (full()).

Si le parking n'est pas rempli, la borne affiche un message (displayPrompt()) et reçoit le fait que l'utilisateur a appuyé sur le bouton (pushButton()), les informations sont envoyées

pour la création d'un ticket (`getInfo()`), ces dernières sont enregistré (`saveData()`) et le ticket est imprimé (`printTicket()`), le client retire le ticket de la borne (`removeTicket()`). La barrière se lève (`BarrierUp()`), le senseur regarde le passage de la voiture (`viewPassage()`) et la barrière se baisse (`BarrierDown()`), le nombre de place de parking est incrémenté de un (`increment()`) et on compare le nombre de place de parking avec le nombre de voitures dans le parking (`compareCapacityFull()`) pour que la méthode `full()` se mette à jour si cette comparaison est vérifiée.

Si le parking est complet, la borne affiche un message (`displayPrompt()`) et rien d'autre n'est effectué.

Si le client a un abonnement, lorsque la borne affiche la demande d'appuie sur le bouton avec `displayPrompt()`, la carte de ce dernier est insérer dans la borne (`insertCard()`). On va alors faire les vérifications dans la classe "subscription" (`verification()`). En interne, on vérifie qu'il rentre un jour de semaine (`verifyDate()`), qu'il soit abonné au parking (`verifyParking()`), que son abonnement ne soit pas expiré (`verifyExpiry()`) et que son abonnement ne soit pas en cours d'utilisation (`verificationUsable()`).

Si toutes les vérifications sont vérifiées, la borne rend la carte à l'abonné (`returnCard()`) et la barrière enregistre l'heure d'entrée (`saveData()`) de l'abonné et la borne rend la carte à celui-ci (`returnCard()`). La barrière se lève (`BarrierUp()`), le senseur regarde le passage de la voiture (`viewPassage()`) et la barrière se baisse (`BarrierDown()`), le nombre de place de parking est incrémenté de un (`increment()`). On ne vérifie pas si c'est complet, car les places qu'utilisent les abonnés n'impacte pas le nombre de places disponible pour les clients ordinaires puisqu'ils ont des places attribuées.

Si les vérifications ne sont pas vérifiées, la borne affiche un message d'erreur (`displayPrompt()`) et la carte de l'abonné lui est rendue (`returnCard()`).

Paielement

Description diagrammes d'activité

L'activité du paiement à la borne se présente comme ceci :

Si la monnaie n'est pas disponible le message "monnaie indisponible serait affiché" .

Le message "Insérer le ticket" s'affiche , le ticket sera inséré dans la borne, le code barre du ticket sera lu, ensuite les informations sur le ticket seront comparées avec les les informations stockées dans le système.

Si la date et l'heure du ticket ne sont pas les mêmes que celles stockées dans le système, le ticket sera rejeté par la borne, et le message "rendez-vous au bureau svp" sera affiché, dans ce cas là on atteint la fin du diagramme des activités. La partie paiement au bureau sera traité dans un diagramme d'activités à part .

Si la date et l'heure du ticket sont les mêmes que celles stockées dans le système, la date et l'heure actuelle sont obtenues, et la durée de séjour dans le parking est calculée, ensuite le montant est calculé et affiché sur l'écran.

Le client va insérer de l'argent, s'il insert un billet un valide, le billet sera éjecté de la fente d'insertion et le message "billet invalide" sera affiché, s'il insert une pièce invalide, la pièce sera rejetée dans le réceptacle de monnaie et le message "pièce invalide" sera affichée, si le billet ou la pièce est valide, le billet ou la pièce sera inséré et sa valeur ajouté au montant total.

Si le montant calculé est inférieur au montant dû le client va encore insérer de l'argent, sinon si le montant calculé est supérieur ou égale au montant dû, le ticket est validé, et la date et heure du paiement pour ce ticket sont ajoutées aux informations stockées sur le système.

Si le montant payé dépasse le montant dû et la monnaie n'est pas disponible, le message "monnaie indisponible" sera affiché , mais si la monnaie est disponible, le montant à rendre est calculé, et il est distribué dans le réceptacle de monnaie, et le message "veuillez récupérer votre monnaie" est affiché, le client récupère sa monnaie, le ticket contenant la date et heure du paiement sera imprimé et éjecter de la fente, ensuite le message "appuyez sur le bouton reçu si vous souhaitez en avoir un" est affiché.

Si le client appuie sur le bouton "reçu" le reçu sera imprimé et éjecté dans le réceptacle à reçus.

Le message "Veuillez prendre le ticket et quitter le parking dans les 15 minutes suivantes" est affiché, le client prend le ticket et le reçu s'il en avait demandé et quitte le parking.

L'activité du paiement au bureau se présente comme ceci :

Le client va donner le ticket rejeté par la borne de paiement à l'employé du parking, qui va vérifier si le ticket a été endommagé, si c'est le cas , l'employé va entrer les informations du ticket dans le système, par la suite le système va comparer les informations entrées par l'employé avec les informations stockées sur le système, et la date et l'heure actuelle sont obtenues et la durée de séjour dans le parking est calculée,

Le montant dû est calculé et affiché sur l'écran, l'employé informe le client sur le montant dû, et le client va donner de l'argent à l'employé qui va prendre l'argent et valide le ticket.

Si le montant payé est supérieur au montant dû, et que la monnaie est disponible, l'employé rend la monnaie au client, sinon si la monnaie n'est pas disponible, le client va juste informer le client qu'il n'y a pas de monnaie.

Dans tous les cas, un nouveau ticket contenant la date et l'heure du paiement est imprimé, l'employé va par la suite donner le ticket au client.

Le système "appuyez sur le bouton 'Reçu' " est affiché sur l'écran de l'employé, l'employé demande au client s'il souhaite en avoir un, si le client dit oui, l'employé va cliquer sur le bouton reçu et le reçu sera imprimé, l'employé donne le ticket et le reçu au client, ou juste le ticket, le client prend son ticket et son reçu ou juste son ticket, et on arrive à la fin de l'activité.

Description diagrammes de séquence

Paieement borne

La séquence du paiement à la borne se présente comme ceci :

Au début de la séquence si la monnaie n'est pas disponible, la borne du paiement va activer l'action récursive « displayPrompt (« Monnaie Disponible ») ».

La borne active la méthode « displayPrompt (« Insert Ticket ») ». Le client va envoyer le message à la borne qui va permettre d'activer la méthode « InsertTicket » de la borne. Par la suite, la borne active sa méthode récursive « readBarCode » et la méthode récursive « CompareInfo(ticket) ».

On a prévu par la suite, un block alternatif, si la date et l'heure sur le ticket sont différentes de celles stockées sur le système, la borne active son action récursive « RejectTicket() », et la méthode « DisplayPrompt (« Rendez-vous au bureau ») », la séquence du paiement au bureau sera traitée dans un diagramme à part.

Si la date et l'heure sur le ticket sont les mêmes que celles qui sont stockées sur le système, la borne active ses méthodes récursives « getCurrentDateHour » « CalculTimeVisit » « CalculPriceParking » « displayPromptPrice ». Ensuite le client va activer la méthode « InsertTicket » de la borne.

On tombe après sur une loop, tant que l'argent inséré est inférieur à l'argent dû, On aura un block alternatif à l'intérieur de la boucle, si le billet n'est pas valide la borne active ses méthodes « rejectBill » et « displayPrompt ». Si la pièce de monnaie n'est pas valide, la borne active ses méthodes « rejectCoin » et « displayPrompt », si l'argent est valide la borne active sa méthode « AddValidCashToTotalCash ».

En sortant de la boucle, la borne active la méthode « validticket » et « savedatapayment ».

On aura un block optionnel, on rentre dedans uniquement si le montant payé est supérieur au montant dû, et on aura un block alternatif à l'intérieur de ce block optionnel, s'il n'y a pas de monnaie, la borne active la méthode « displayprompt « aucune monnaie n'est rendu » », sinon si la monnaie est disponible, la borne active ses méthodes « CalculReturnCash » et « returnCash » et « displayPrompt 'Récupérer votre monnaie' ».

Ensuite, la borne active sa méthode « printTicket(Date,Hour) », et « displayPrompt'Receive' », et on aura un autre block optionnel si le client clique le bouton receipt, la borne active sa méthode « printReceipt ».

A la fin de la séquence, la borne active sa méthode « displayPrompt'prendreticket' ».

Paielement bureau

La séquence du paiement au bureau se présente comme ceci :

Au début de la séquence, l'utilisateur active sa méthode « GiveRejectedTicket() » et le résultat sera renvoyée à la borne, l'employé active sa méthode « VerifyTicket() », si le ticket est endommagé l'employé active sa méthode « EnterTicketInformation() », et la valeur sera renvoyée au système, qui va par la suite activer ses méthodes récursives CompareInformation()

, GetCurrentTime(), CalculateLengthOfStay(), CalculateAndDisplayAmountOwed(), et le résultat sera renvoyée à l'employé qui va activer sa méthode InformClient(), et la valeur sera renvoyée au client, le client active sa méthode GiveCash(), et le résultat est renvoyé à l'employé, qui active ses méthodes TakeCash(), ValidTicket(), AddPaymentTime() , et le résultat est renvoyé au système.

On aura après un block optionnel, si le montant payé est supérieur au montant dû, A l'intérieur de ce block optionnel il y a un block alternatif, si la monnaie n'est pas disponible, l'employé active sa méthode InformClient(), et la valeur est renvoyée au client, si la monnaie est disponible , l'employé active sa méthode GiveChange() et le résultat sera renvoyé au client qui lui active sa méthode TakeChange(), en sortant du block alternatif, le système active sa méthode PrintTicket()

, et la valeur est renvoyée à l'employé qui lui va activer sa méthode AskClient(), et le résultat est renvoyé au client.

On tombe sur un block alternatif, si le client dit non, l'employé active sa méthode GiveTicket()

, et renvoie le résultat au client, sinon si le client dit oui , le client active sa méthode PressOnReceiveButton() et le résultat est renvoyée au système, qui par la suite active sa méthode PrintReceive(), et le résultat est à nouveau renvoyée à l'employé qui active sa méthode GiveTicketAndReceive(), et le résultat est renvoyée au client.

En sortant du block alternatif, le client active sa méthode TakeTicketOrTicketAndReceive

(.). Et la séquence sera finie.

Quitter le parking

Description diagramme d'activité

L'activité de la sortie du parking se présente comme ceci :

La voiture roule jusqu'au senseur, puis lorsque le senseur détecte la voiture, le message « Insérer le ticket » s'affiche sur l'écran de la borne, le client va insérer le ticket. Dans le cas où c'est un client abonné la barrière sera levée et l'heure de sortie du client va être enregistrée, par contre si c'est un client ordinaire, le barreau sera lu et par la suite on aura deux cas de figure, si moins de 15 sont passées après le paiement la barrière se lèverait et l'heure de sortie du client sera enregistrée sinon si plus de 15 minutes sont passées l'interphone sera activée pour permettre au client de parler à l'employé, l'employé va activer la barre à distance depuis le système et il va enregistrer l'heure de sortie du client dans le système.

La voiture va ensuite rouler jusqu'au deuxième senseur, et quand elle sera détectée, la barrière sera baissée.

Si le parking n'était pas plein on serait à la fin de l'activité de la sortie du parking, sinon si le parking était plein, le nombre de véhicules serait décrémenté de 1, et le message 'places disponibles sera affiché à l'entrée.

S'il n'y a pas de véhicules en attente on sera à la fin de l'activité, sinon la borne du véhicule qui attend sera notifiée, si le client n'appuie pas sur le bouton, la borne sera notifiée pour le prochain véhicule, on aurait une boucle jusqu'à ce qu'il y ait plus de véhicules en attente, et on va atteindre la fin de l'activité. L'activité "voir statut borne" se fait en parallèle avec toutes les autres activités et le statut sera mis à jour chaque 10 secondes.

Description diagramme de séquence

La séquence de la sortie du parking se présente comme ceci :

Si le client choisit de voir le statut de la borne il va activer la méthode (SeeStatutBorne()) de la borne avec une condition de garde de mise à jour chaque les 10 secondes.

Le senseur ("sensor") détecte le client et demande à la borne si la sortie est possible (askExit()). La méthode (displayPrompt()) de la borne sera activée.

On aura un block alternatif, si c'est un client abonné, il va activer la méthode « InsertCard » de la borne, la borne va activer la méthode « BarrierUp » de la barrière et la barrière active la méthode viewPassage() du senseur, dès que le senseur détecte le passage du véhicule, il envoie le message à la barrière pour activer la méthode « BarrierDown » et même temps il envoie le message au système pour permettre le sauvegarde de l'heure de sortie du véhicule avec la méthode « saveExitHour ». Ensuite le message sera envoyé au Parking pour décrémenter le nombre de véhicules « decrement(nbCar) » et comparer ce nombre avec la capacité du parking « compareCapacityFull(nbCar,nbSpace) ».

Dans le même block alternatif, si c'est un client ordinaire il va activer la méthode « InsertTicket » de la borne, la borne active l'action récursive « readBarCode » pour lire le code barre du ticket.

On rentrera dans un block alternatif, si moins de 15 minutes sont passées, la borne va activer la méthode « BarrierUp » de la barrière et la barrière active la méthode viewPassage() du senseur, dès que le senseur détecte le passage du véhicule, il envoie le

message à la barrière pour activer la méthode « BarrierDown » et même temps il envoie le message au système pour permettre le sauvegarde de l'heure de sortie du véhicule avec la méthode « saveExitHour ». Ensuite le message sera envoyé au Parking pour décrémenter le nombre de véhicules « decrement(nbCar) » et comparer ce nombre avec la capacité du parking « compareCapacityFull(nbCar,nbSpace) ».

Si plus de 15 minutes sont passées, la borne active l'action récursive « useInterCom » qui permet d'activer l'interphone relié au bureau de l'employé, l'employé va ensuite activer la méthode « BarrierUpDistance » de la barrière, et la barrière active la méthode viewPassage() du capteur, dès que le capteur détecte le passage du véhicule, il envoie le message à la barrière pour activer la méthode « BarrierDown » et même temps le résultat sera renvoyé au système pour qu'il enregistre l'heure de sortie du véhicule avec la méthode « saveExitHour », Ensuite le message sera envoyé au Parking pour décrémenter le nombre de véhicules « decrement(nbCar) » et comparer ce nombre avec la capacité du parking « compareCapacityFull(nbCar,nbSpace) ».

A la fin, on aura un block optionnel si le parking était plein, la borne active la méthode récursive qui permet d'afficher le message places sur la borne de l'entrée avec la méthode « displayPrompt 'places' » (borne d'entrée et borne de sortie sont des sous-classes de la classe borne) , à l'intérieur de ce block optionnel on aura une boucle tant que le client n'appuie pas sur le bouton on continue de notifier la borne de contrôle.

Enregistrement de visite de sécurité

Description diagramme d'activité

La municipalité signe un contrat avec certaines sociétés de sécurité. Ces sociétés envoient des gardes pour visiter et inspecter le parking. Chaque visite est enregistrée, ce diagramme d'activité commence par montrer l'accord passé entre deux parties, la société de sécurité et la municipalité. L'entreprise envoie des gardes sur le parking selon les termes du contrat. Il arrive au parking, s'il a besoin de garer sa voiture, il peut utiliser la carte qui lui a été remise par l'entreprise pour accéder au parking. Le garde entre ensuite dans le bureau où il valide sa carte, c'est cette même carte qui peut être utilisée pour entrer dans le parking comme n'importe quel autre client. Le valideur de tickets du bureau scanne sa carte et enregistre la date et l'heure de la visite. La durée de la visite est précisée dans le contrat.

Description diagramme de séquence

Cette partie du parking se compose de quatre classes principales : municipality, company, guard et TicketValidator. La municipalité signe un contrat avec les sociétés de sécurité, de ce fait, lorsque la municipalité fait appel à un garde de l'une des sociétés de sécurité, la méthode "sendGuard" est appelée à partir de la classe company. La méthode de "visitParking" est invoquée par la classe du guard, qui fait appel à la méthode arrive. Comme la procédure d'entrée est détaillée dans la partie "entrée du parking" de ce rapport, nous n'avons pas à le présenter ici. La méthode park résume l'entrée dans le parking, ainsi que le stationnement de la voiture. Nous avons également utilisé "opt" pour souligner la seule condition que invoque la méthode park, "driving". Après l'invocation de la méthode park si la condition était vraie ou après l'invocation de la méthode de l'arrivée si la condition était fausse, la méthode enterOffice est invoquée, ensuite InsertCard. La méthode recordVisit de la classe TicketValidator est invoquée, enfin à la fin de la visite, la méthode leave est appelée.

Enregistrement de fautes & Accord de niveau de service surveillance

Description diagramme d'activité

Comme l'indique son nom "diagramme d'activité", le diagramme élucide toutes activités pouvant faire l'objet de notre projet , dans ce cas nous dirons de cette sous-partie du projet. Elle dessine le cheminement de manière explicite de toutes les étapes de la partie concernée du projet. A titre illustratif, nous pourrions parler de notre partie qui se présente comme suit:

Tout d'abord, nous avons l'arrivée de l'utilisateur qui se traduit par la présentation d'une carte d'abonnement ou d'un ticket, une fois celui validé, la barrière est automatiquement levée sinon, il reste baissé. Quant au système du parking , lui, il détecte les fautes s'il en y a.

Dans cette perspective, nous avons utilisé les "opt" pour poser ou parler de la condition.

Les fautes en fonction de leurs natures , une fois détectées sont automatiquement envoyées au système d'enregistrement du parking et parallèlement une notification est également envoyée aux entreprises de maintenances adéquates. Le service de maintenance notifié envoie à son tour une demande rapport de fautes au système afin de procéder à la réparation.

Description diagramme de séquence

L'utilisateur insère la carte dans la borne, chose qui crée une interaction entre la borne elle-même et le système du parking, bien qu'elle fasse partie du système du parking. Suite à la vérification de la validité du ticket ou de l'abonnement, la barrière est levée. Par ailleurs, lorsque des fautes sont détectées par le système, celles-ci sont retournées et enregistrées dans le système d'enregistrement de fautes y compris la date et l'heure auxquelles elles ont eu lieu. Lorsque des fautes sont détectées, selon leurs natures, des notifications sont générées et transmises aux entreprises adéquates de maintenance qui sont censées faire les réparations.

Aussi, un rapport mensuel de statistiques de toutes les fautes doit pouvoir être produit en cas de demande de la part de l'entreprise.

Abonnements

Description diagramme d'activité

Le diagramme d'activité ci-dessus représente le flux d'une activité à une autre dans notre système d'abonnement. Le client demande un abonnement, l'administrateur vérifie si le client est déjà membre, donc le nœud de décision. Si le client signe pour la première fois, il sera inscrit et ses données personnelles seront enregistrées dans le système. S'il s'agit d'une demande de renouvellement, le système renouvellera le client en fonction de la date d'expiration. La nouvelle date d'expiration est fixée à partir de la date actuelle si la carte a déjà expiré, sinon, elle sera prolongée à partir de la date d'expiration indiquée. Les attributs sont mis à jour dans le système. Il est important de sauvegarder les données et de mettre à jour les informations car ces informations sont nécessaires à la rédaction du rapport de fin d'année : le système calcule le prix, l'affiche et le client paie. Si le paiement est accepté, alors le processus a été un succès, le client reçoit une carte d'abonnement ou sa date d'expiration est réinitialisée, si le paiement n'est pas accepté, le processus est annulé.

Description diagramme de séquence

Dans le diagramme de séquence de l'abonnement disponible en annexe. Nous avons modélisé un seul système pour l'abonnement et le renouvellement de l'abonnement.

Tout d'abord, l'abonné demande l'adhésion, bien qu'il n'utilise pas le système pour faire cette demande, il est important de montrer que la demande déclenche l'interaction entre l'administrateur et le système d'abonnement, la demande invoque la méthode "verify client" de la classe d'administration. Cela signifie que lorsque l'administrateur reçoit une demande, il vérifie si la personne qui fait la demande est déjà membre ou si elle fait une demande pour la première fois. Le premier "alt" décrit deux scénarios possibles, si la condition de "nouveau membre" est vraie, alors la demande est faite pour la première fois (demande d'abonnement), la méthode "sign up a member" de la classe d'administration sera invoquée.

Comme cela implique la création d'un nouveau profil, des informations personnelles telles que le nom, le prénom, le numéro de téléphone, l'adresse, etc. seront saisies dans le système. Le système crée alors un membre et enregistre les données saisies par l'administrateur. Toutefois, si la condition est fausse, la méthode "renouveler un membre" sera invoquée par la classe d'administrateur, qui vérifie à son tour deux conditions, démontrées par le deuxième "alt".

Si la condition expirée est vraie, alors la méthode "set new date (from the current date)" du système est invoquée, sinon, la nouvelle date d'expiration sera reportée à partir de la date d'expiration. Lors du renouvellement de l'adhésion, le système met également à jour la date. Après la création et le renouvellement de l'adhésion, la méthode "display payment" sera invoquée, cette dernière invoque la méthode "ask for payment" de la classe d'administration. Le client paie, puis le système imprime la carte d'abonnement. S'il s'agit d'un client qui renouvelle son abonnement, la date d'expiration est prolongée et il conserve la même carte qu'il aurait reçue lors de son premier abonnement.

Autres requêtes de gestion

Description diagramme d'activité

L'activité des autres requêtes de gestion se présente comme ceci :

S'il y a des abonnés, le système vérifie les dates de fin d'abonnement ("Verify ends of the subscription's date"), si la fin de l'abonnement est supérieur à un an, on supprime l'abonnement ("Delete subscription") sinon on ne supprime pas l'abonnement.

Pour la révision annuelle des tarifs, on affiche le revenu ("Display income") puis le système estime le niveau de revenue de l'année en cours ("System estimate une income for the current year") et l'affiche ("Display the estimate to the municipality"). Tant que l'estimation avec le nouveau prix ne correspond pas aux attentes de la municipalité, on teste l'estimation avec un nouveau prix ("Test the estimation with the new prices"). Lorsque le nouveau prix leur convient, les tarifs sont mis à jour.

Description diagramme de séquence

La séquence des autres requêtes de gestion se présente comme ceci :

S'il y a des abonnés, le système vérifie les dates de fin d'abonnement (`verifySubscription()`), si la fin de l'abonnement est supérieur à un an par rapport à la date du jour (`if CurrentDate >= expiryDate + 1 year`), on supprime l'abonnement ("Delete subscription") sinon on ne supprime pas l'abonnement. On fait cela pour tous les abonnements.

Pour la révision annuelle des tarifs, la municipalité (`municipality`) demande au système (`system`) le revenu annuelle (`askYearIncome()`), le système calcule le revenu (`CalculateIncome()`) puis affiche le revenu (`displayIncome()`). Tant que l'estimation avec le nouveau tarif ne correspond pas aux attentes de la municipalité (`while TryPrices(NewPrice) != what the municipality expect`), on teste l'estimation avec un nouveau tarif (`TryPrices()`), le système calcule le revenu avec le nouveau tarif (`CalculateIncome()`) et affiche le résultat du test du nouveau prix (`displayTest()`). Lorsque le nouveau prix leur convient, les tarifs sont mis à jour (`saveChangePrice()`).

Diagramme de classes

Nous avons plusieurs relations de classes, principalement des associations et peu de relations.

La borne utilise 4 senseurs (sensor), 2 à l'entrée et 2 à la sortie, elle vérifie zéro ou un abonnement (subscription), active également 2 barrières (barrier) et elle imprime zéro ou un ticket (ticket).

Le client insère sa carte ou son ticket dans la borne. Le senseur détecte zéro ou plusieurs clients et le client est détecté par 4 senseurs, 2 à l'entrée et 2 à la sortie. Le client insère sa carte dans la borne (ticket). Le système d'abonnement est composé de client qui s'abonne et d'un rapport annuel (report). Le client s'abonne grâce à l'administrateur. Le client peut demander de l'aide à zéro ou un employé et un employé peut aider zéro ou plusieurs clients. Le client peut utiliser une ou plusieurs borne de paiement et la borne peut être utilisée par un client à la fois.

Le système peut être réparé par une ou plusieurs compagnies et la compagnie peut réparer un système à la fois. Le système peut être vérifié par un seul "ticketValidator" et le "ticketValidator" peut vérifier un seul système. Le système peut être utilisé par un ou plusieurs employés et l'employé peut utiliser un seul système.

Les compagnies de maintenance sont composées d'un moyen de réparation, ils peuvent donc faire zéro ou plusieurs réparations. Le système d'enregistrement des fautes prévient une ou plusieurs compagnies de maintenance selon le problème. Le système d'enregistrement de fautes est composé de zéro ou plusieurs fautes.

La municipalité peut interagir avec le système d'abonnement et elle signe un contract avec une ou plusieurs compagnies de sécurité.

Le parking peut avoir deux barrières, une à l'entrée et une à la sortie pour chaque parking. L'administrateur peut inscrire zéro ou plusieurs clients et le client peut s'inscrire avec un ou plusieurs administrateurs. Le garde enregistre sa visite avec la borne du bureau.

Uses-cases

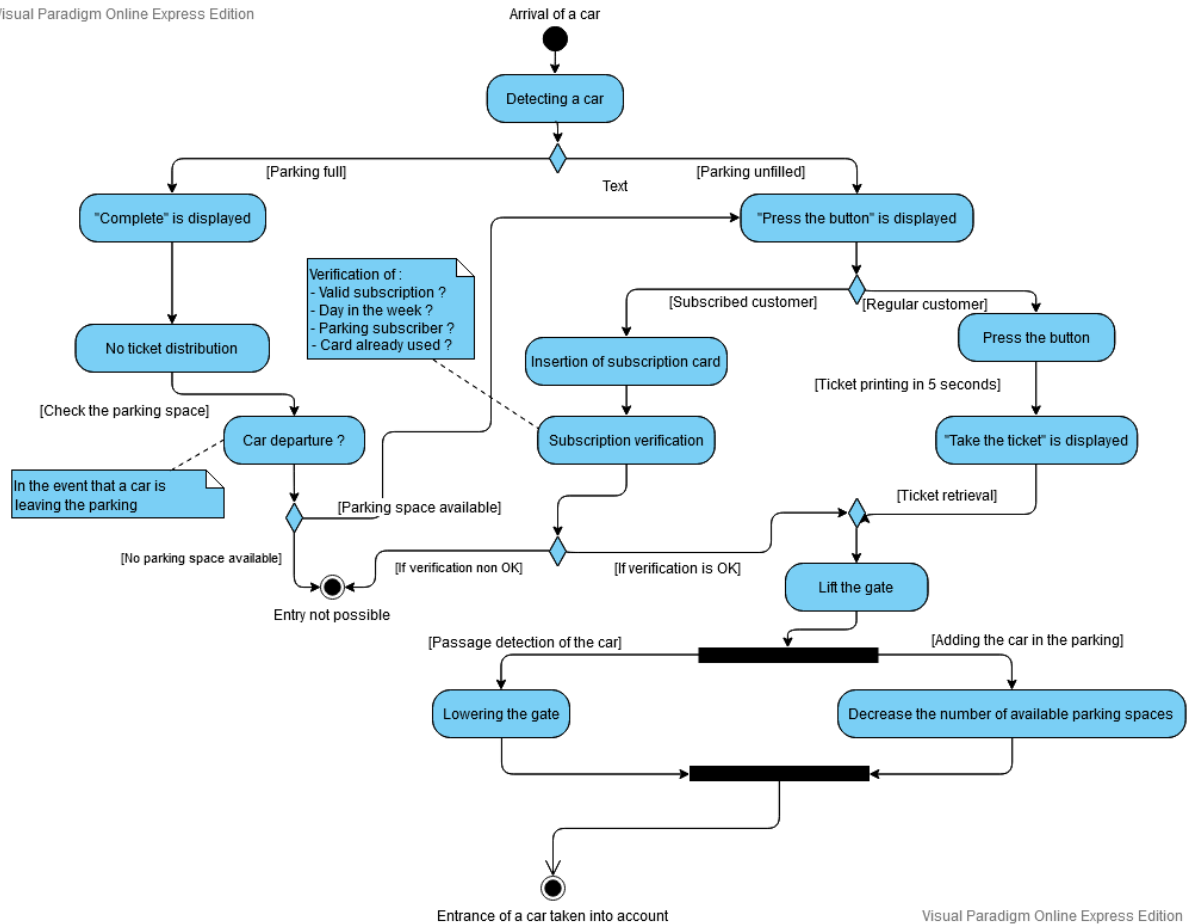
Le système peut être utilisé par plusieurs acteurs :

- L'employé a plusieurs cas d'utilisations différentes, comme par exemple inscrire un membre pour le lien avec l'abonnement et peut également ouvrir une barrière à distance en cas de problèmes.
- La municipalité peut voir les affichages du système, faire des demandes aux systèmes ainsi que tester des nouveaux prix
- La borne a plusieurs cas d'utilisations, comme par exemple afficher des messages, imprimer des tickets et vérifie les abonnements
- Le client a également plusieurs cas d'utilisations comme les cas suivants. Il peut communiquer via l'interphone avec l'employé, insérer un ticket ou une carte et s'abonner au parking.
- La compagnie de maintenance reçoit la notification des fautes.
- Le système peut avoir comme cas d'utilisation, la vérification des abonnements, l'affichage des messages et le calcul des revenus à partir du rapport annuel.

Annexe

Diagramme

Visual Paradigm Online Express Edition



Visual Paradigm Online Express Edition

Diagramme d'activité : Entrée Parking

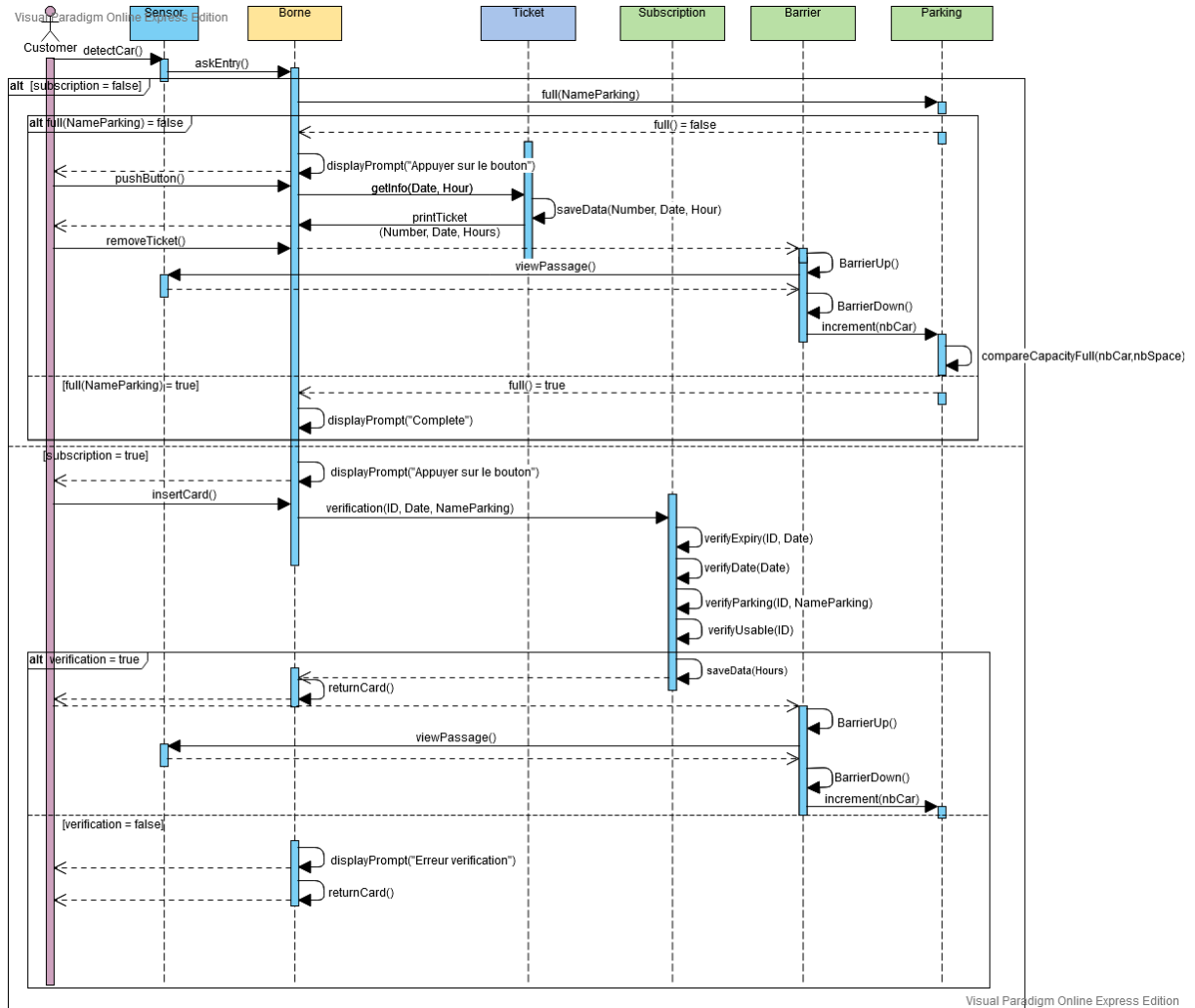
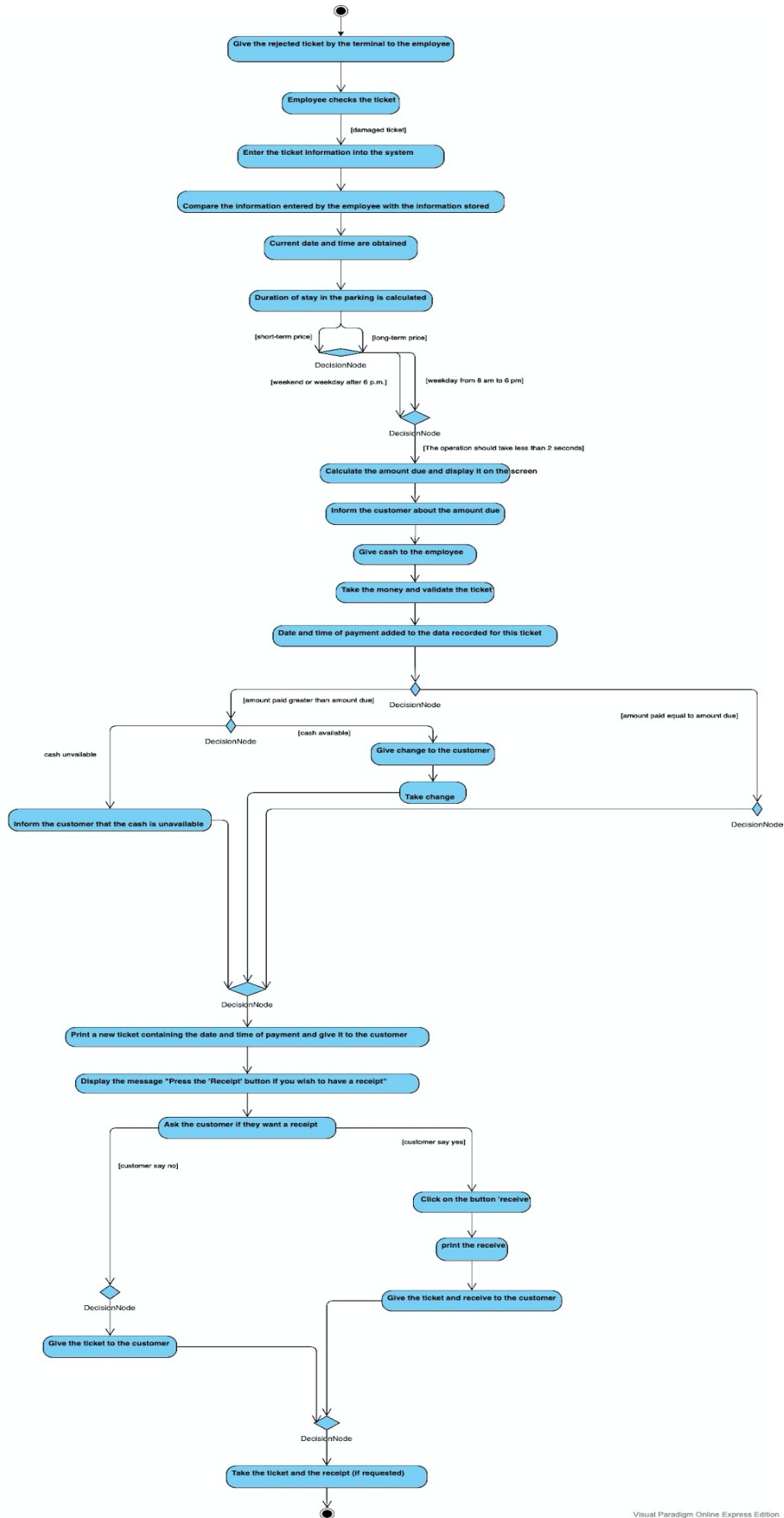


Diagramme de séquence : Entrée Parking



Visual Paradigm Online Express Edition

Diagramme d'activités : Paiement au bureau

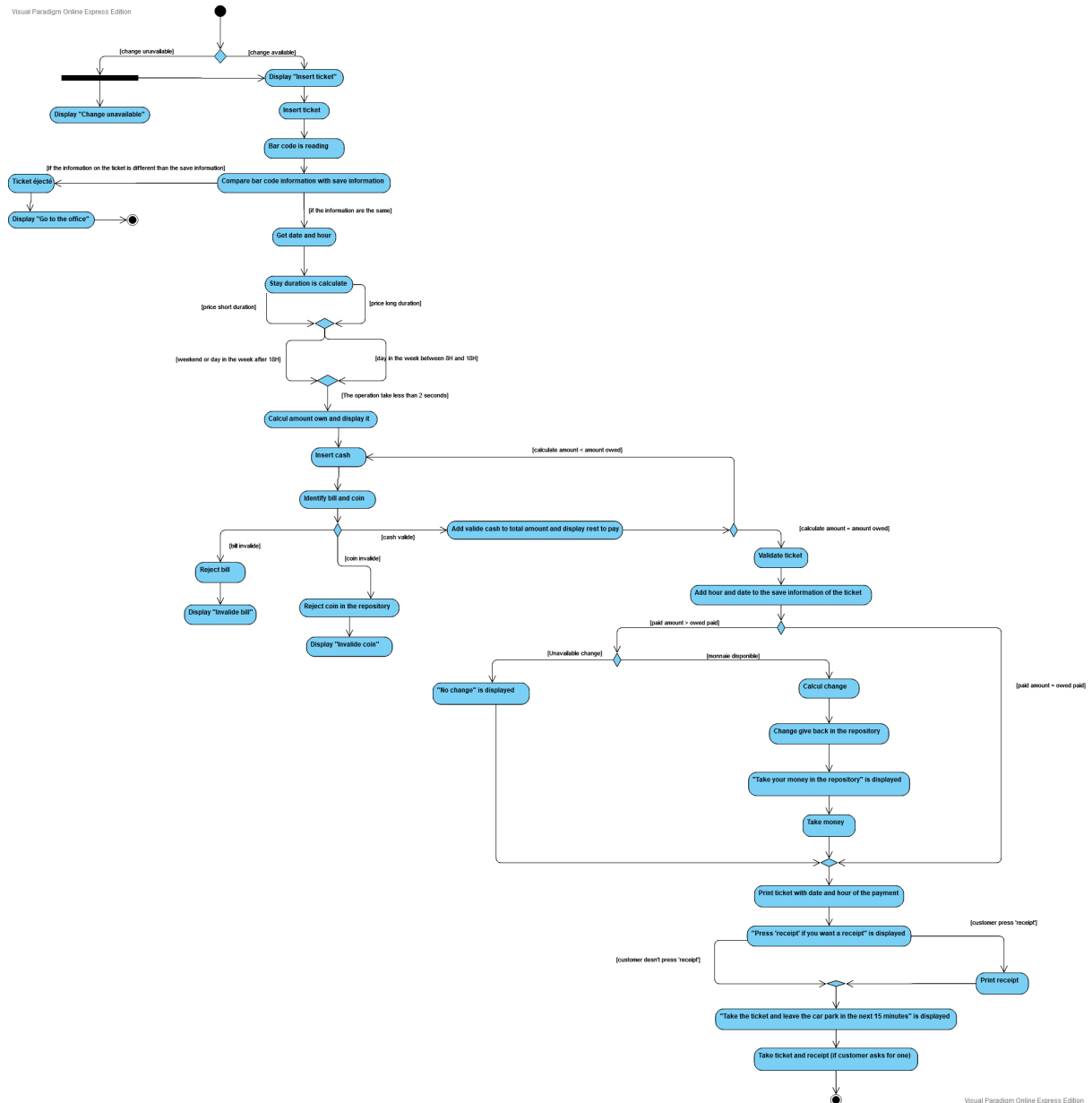


Diagramme d'activité : Paiement à la borne

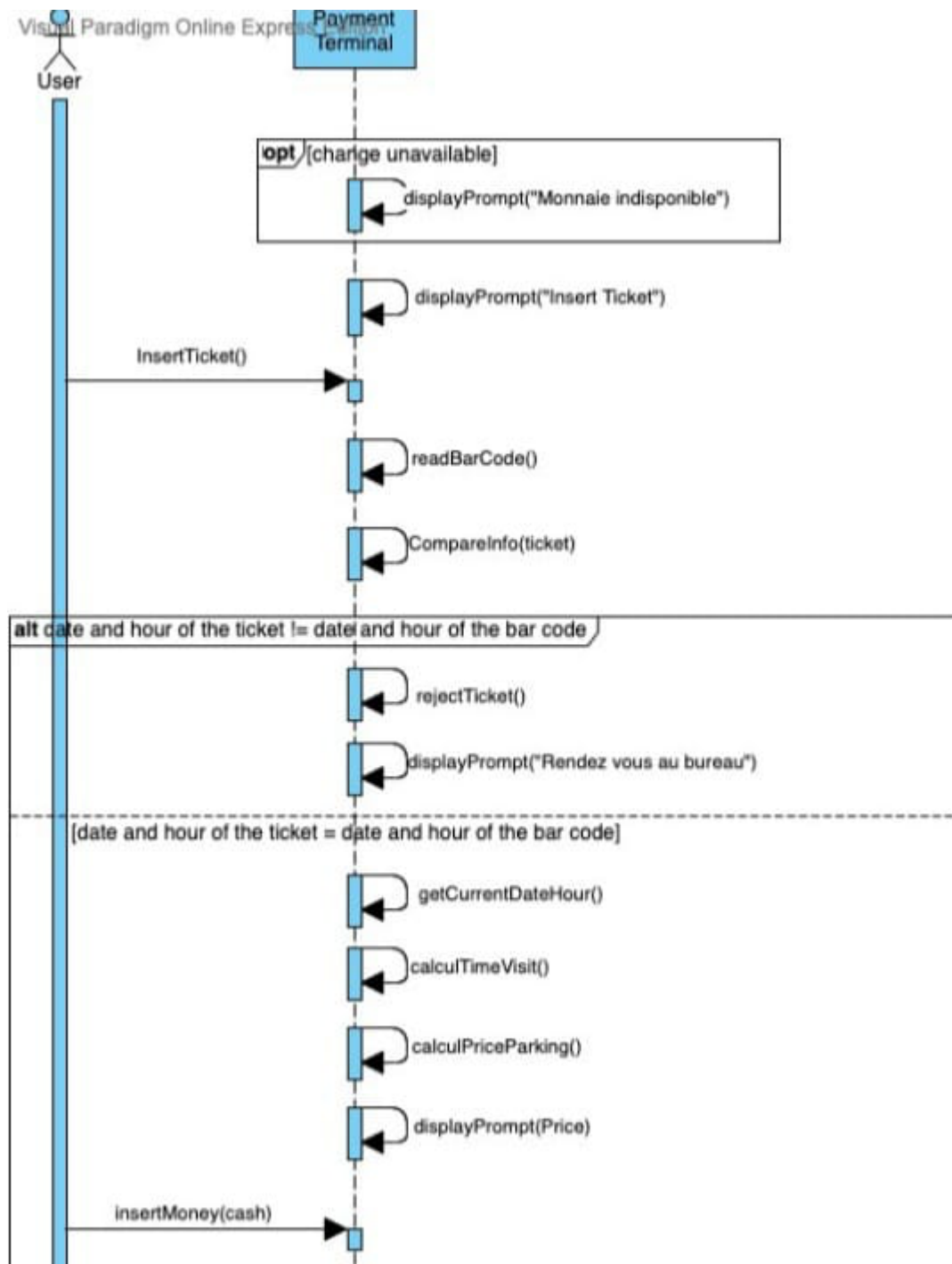


Diagramme de séquence : Paiement à la borne (début diagramme)

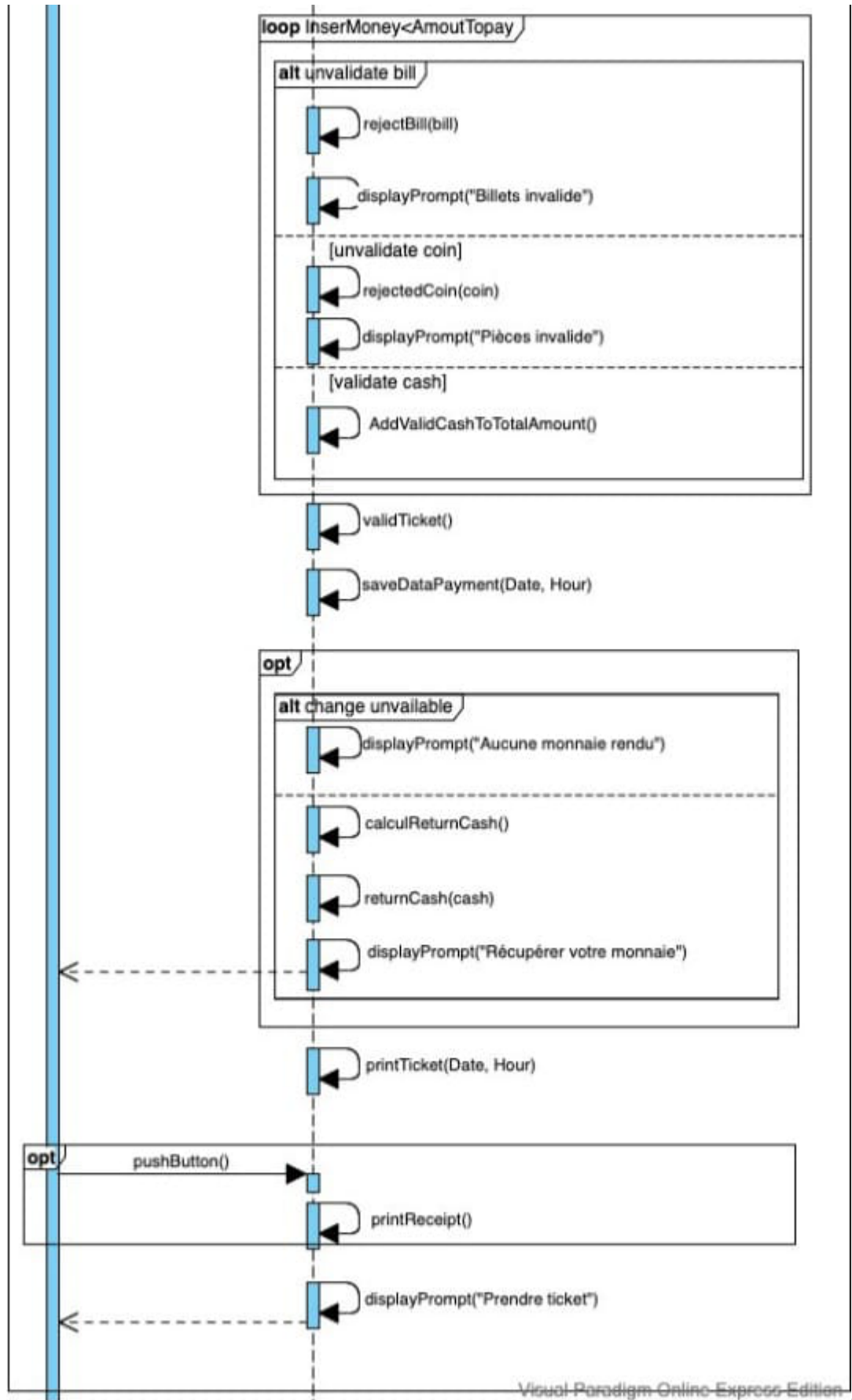
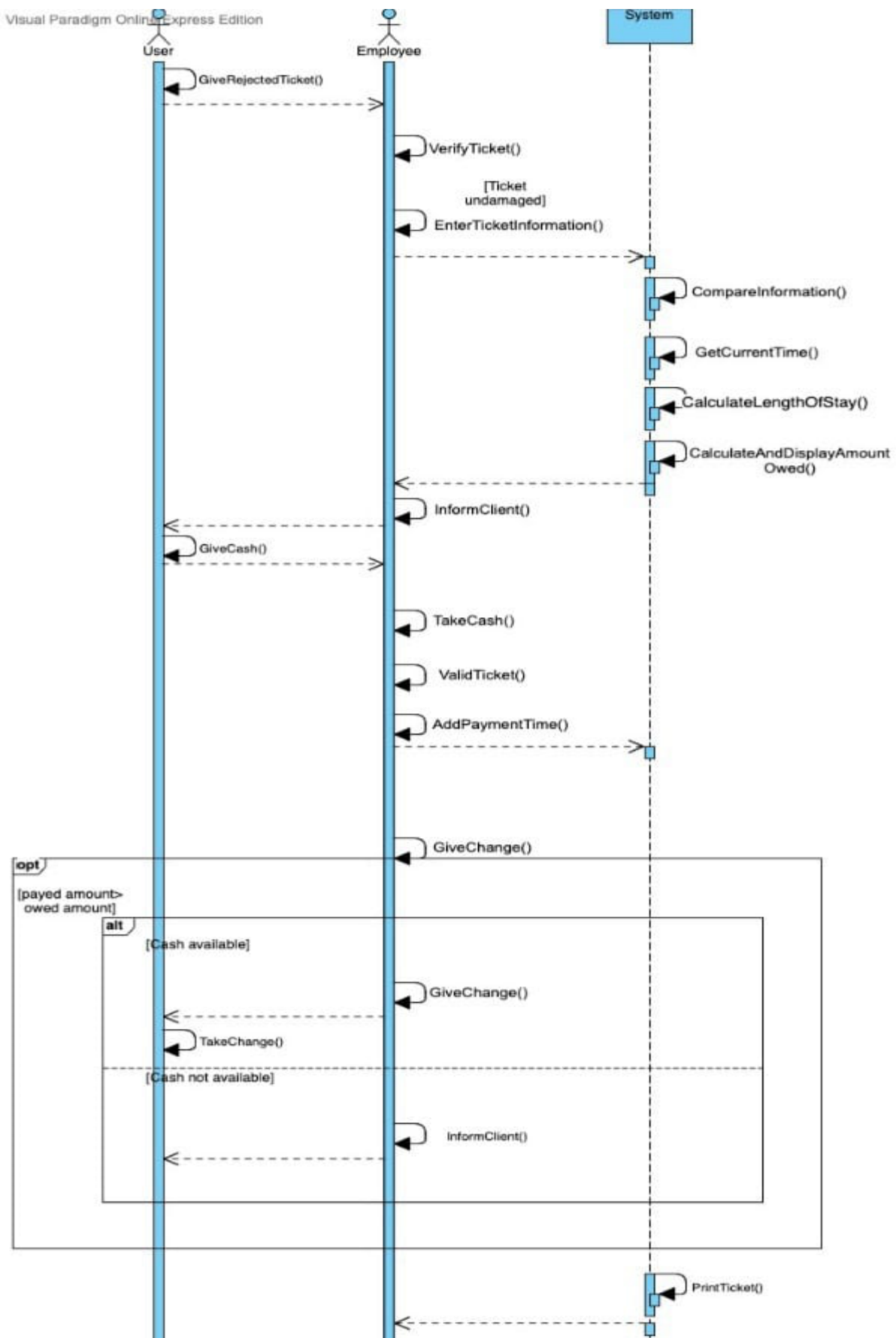
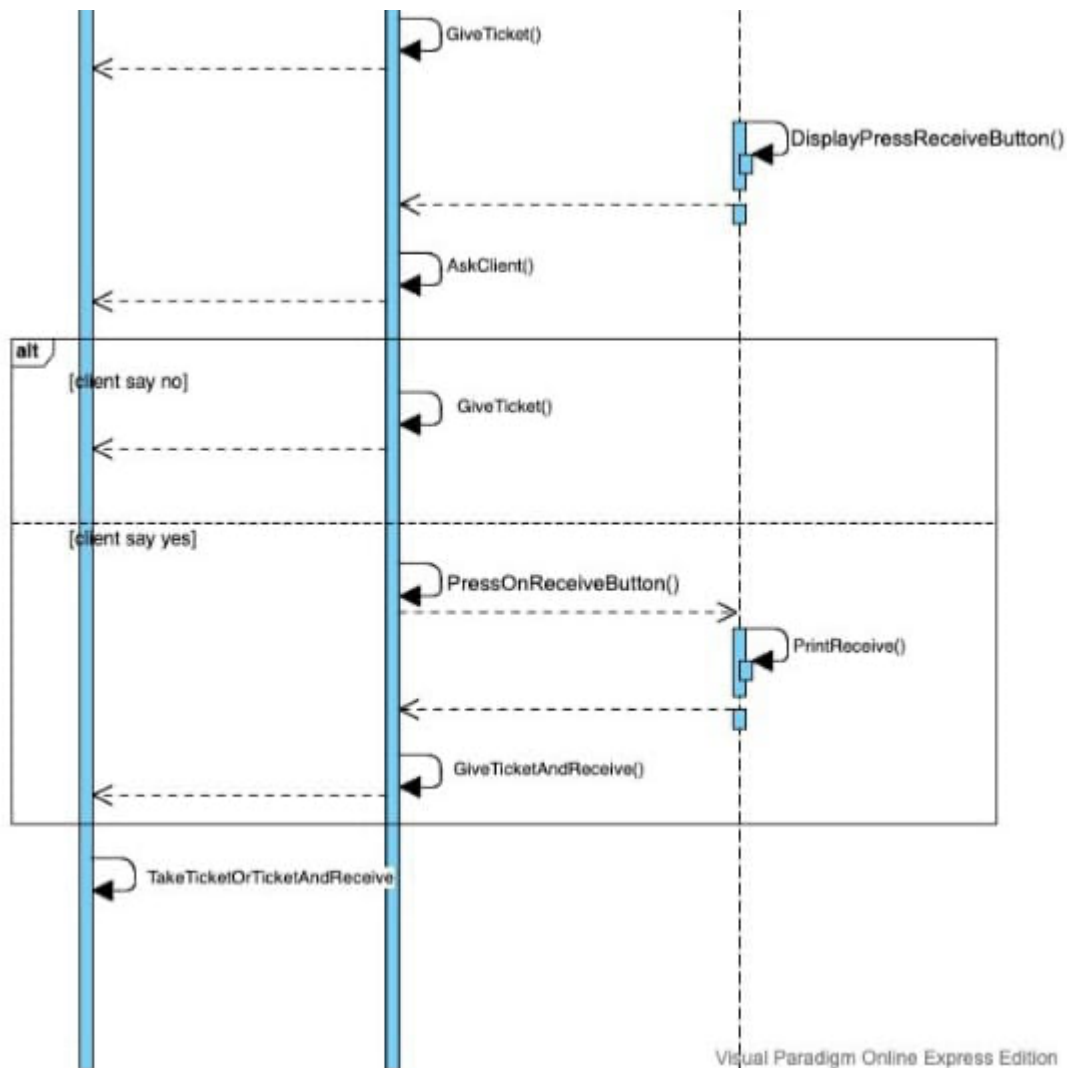


Diagramme de séquence : Paiement à la borne (suite du diagramme)





Visual Paradigm Online Express Edition

Diagramme de séquence : Paiement au bureau

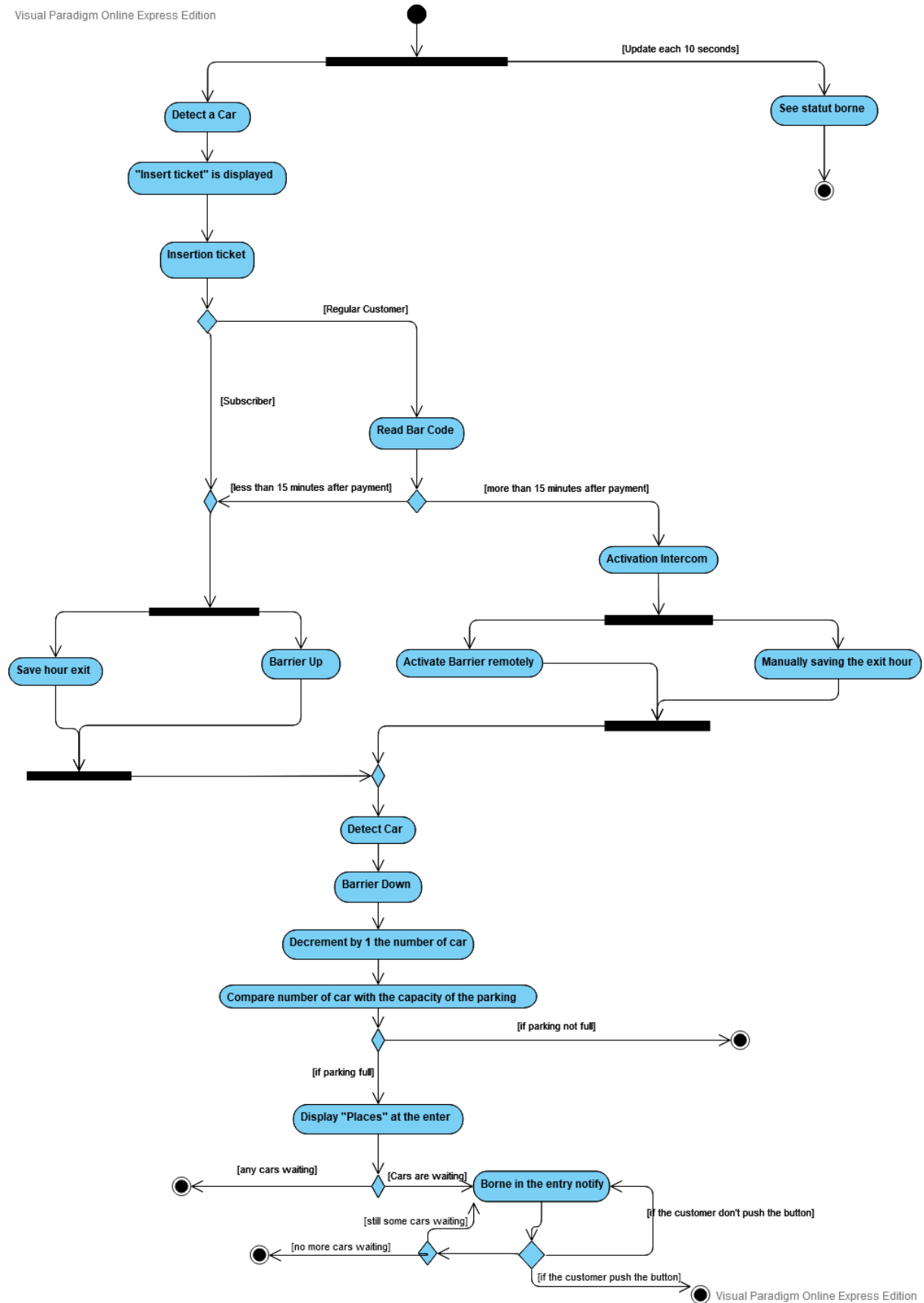


Diagramme d'activité : Quitter le parking

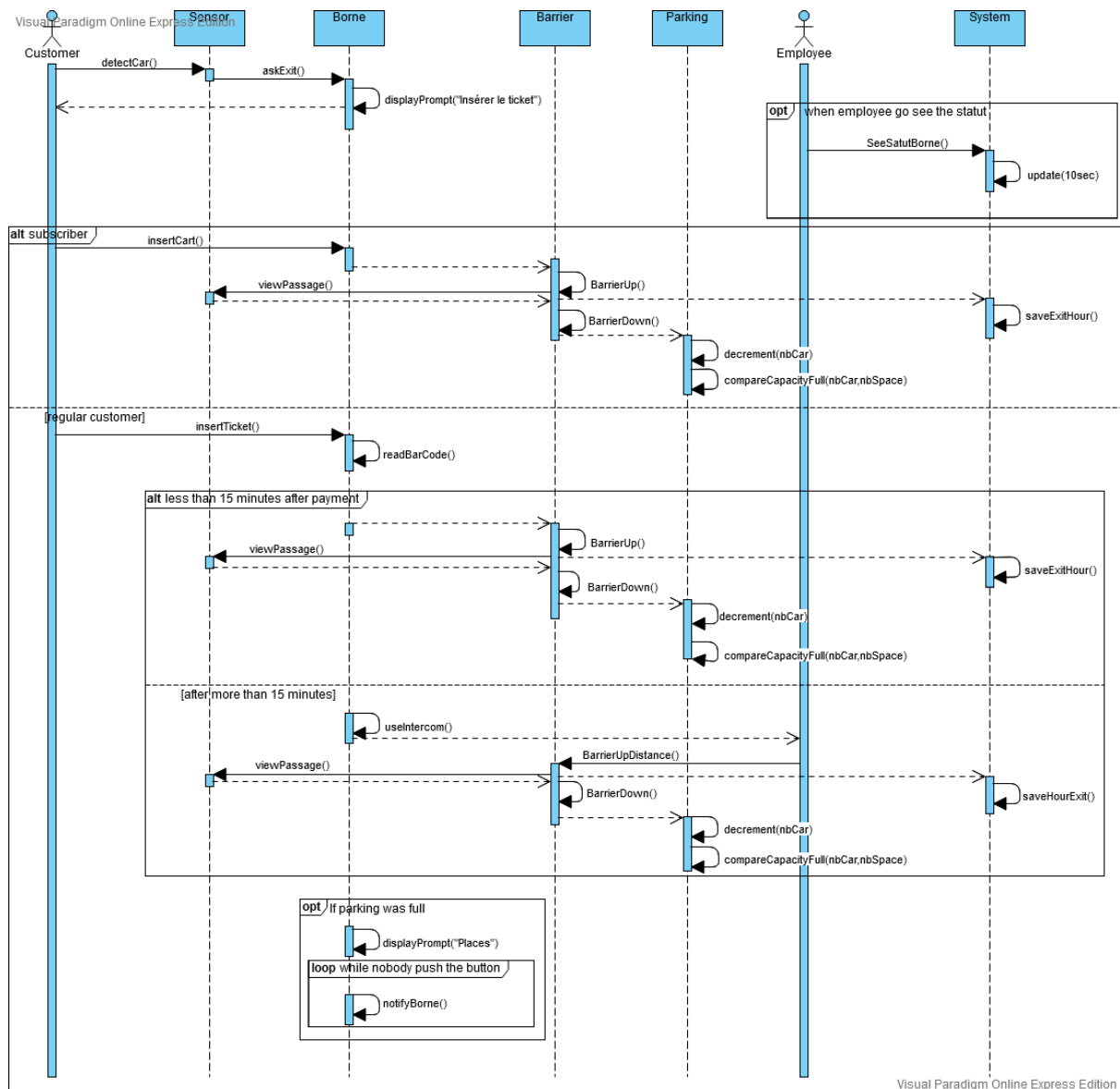
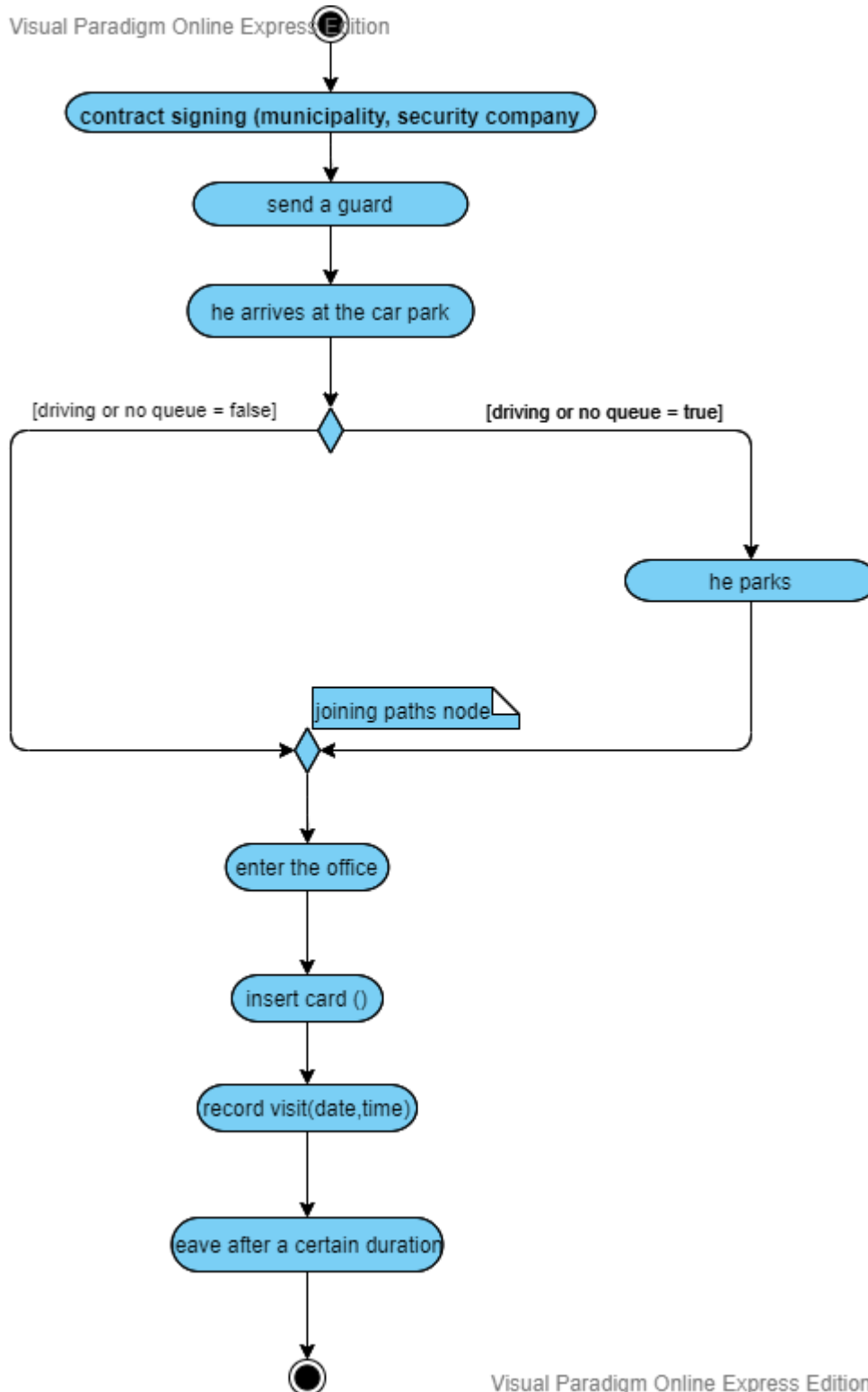


Diagramme de séquence : Quitter le parking



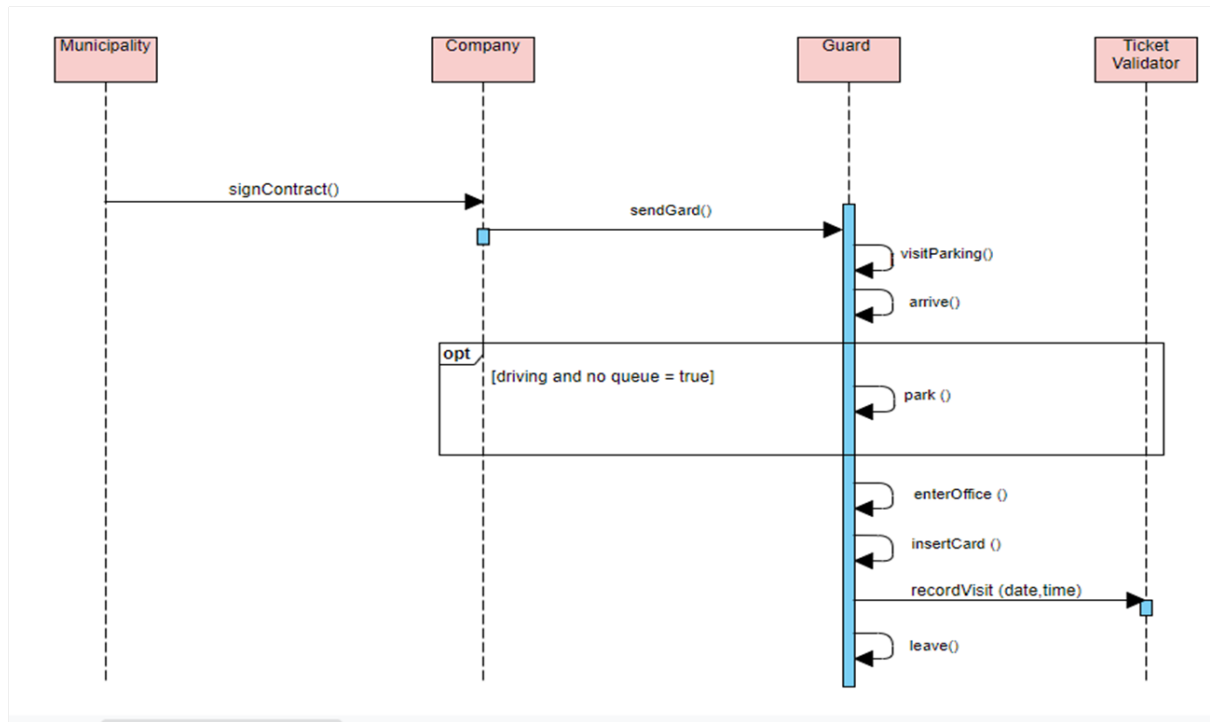


Diagramme de séquence: enregistrement de visite de sécurité

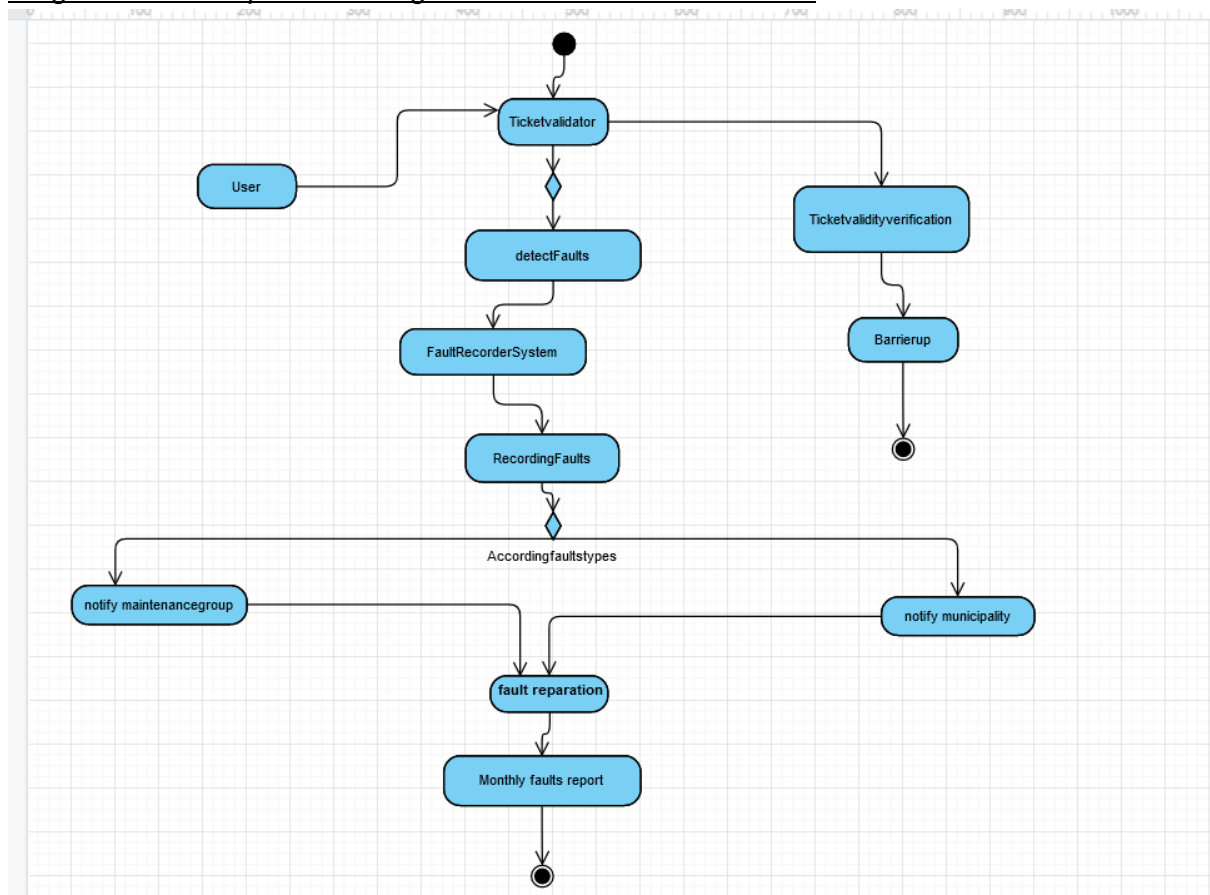


Diagramme d'activité : Enregistrement des fautes et accord de niveau de service de surveillance

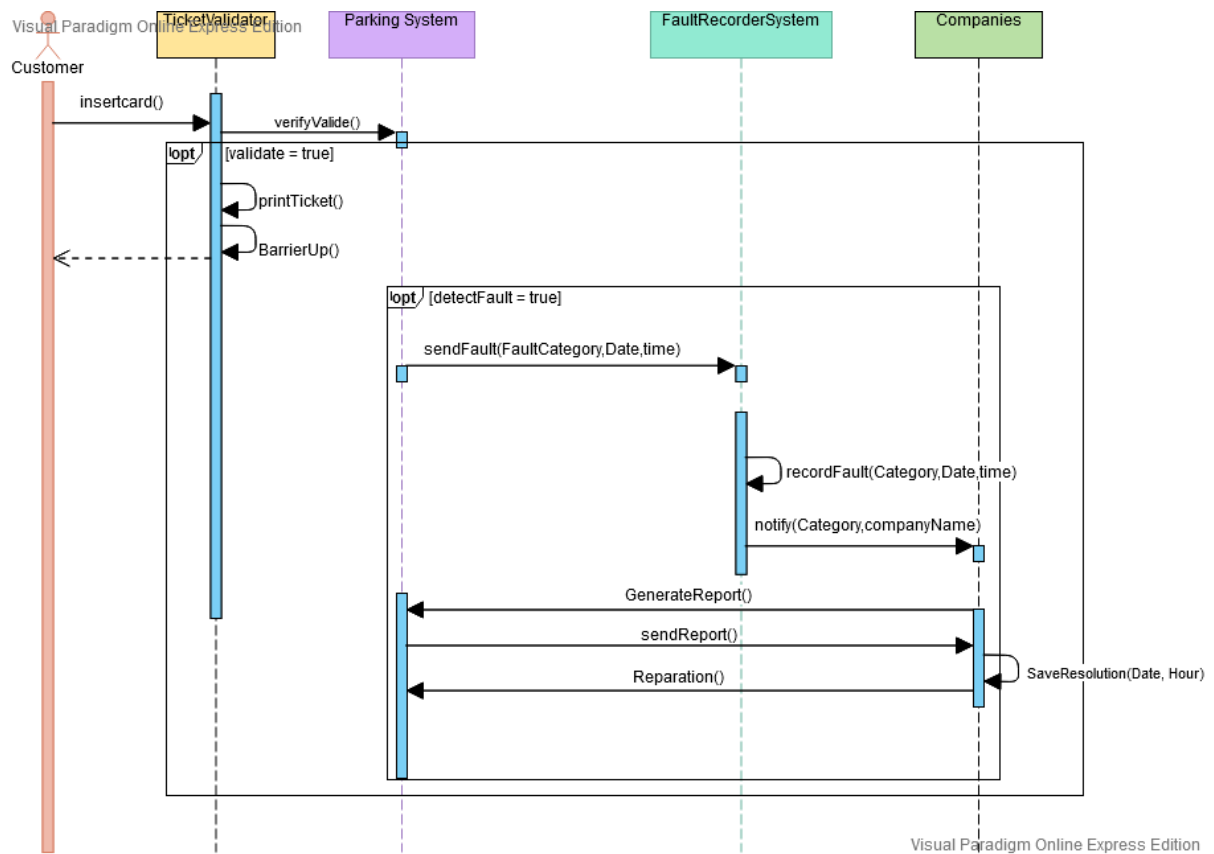


Diagramme de séquence : Enregistrement des fautes et accord de niveau de service de surveillance

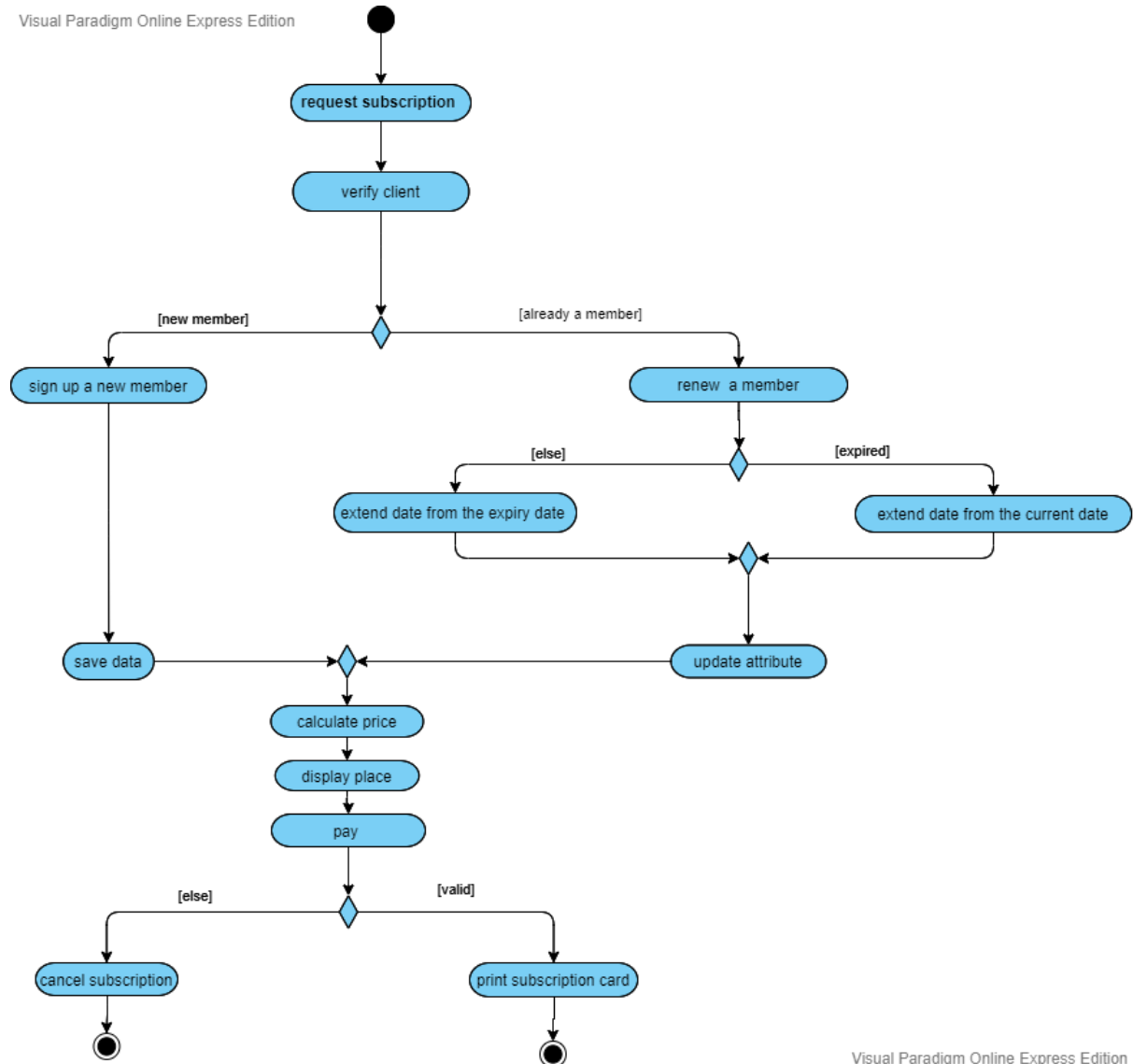


Diagramme d'activité : Abonnement

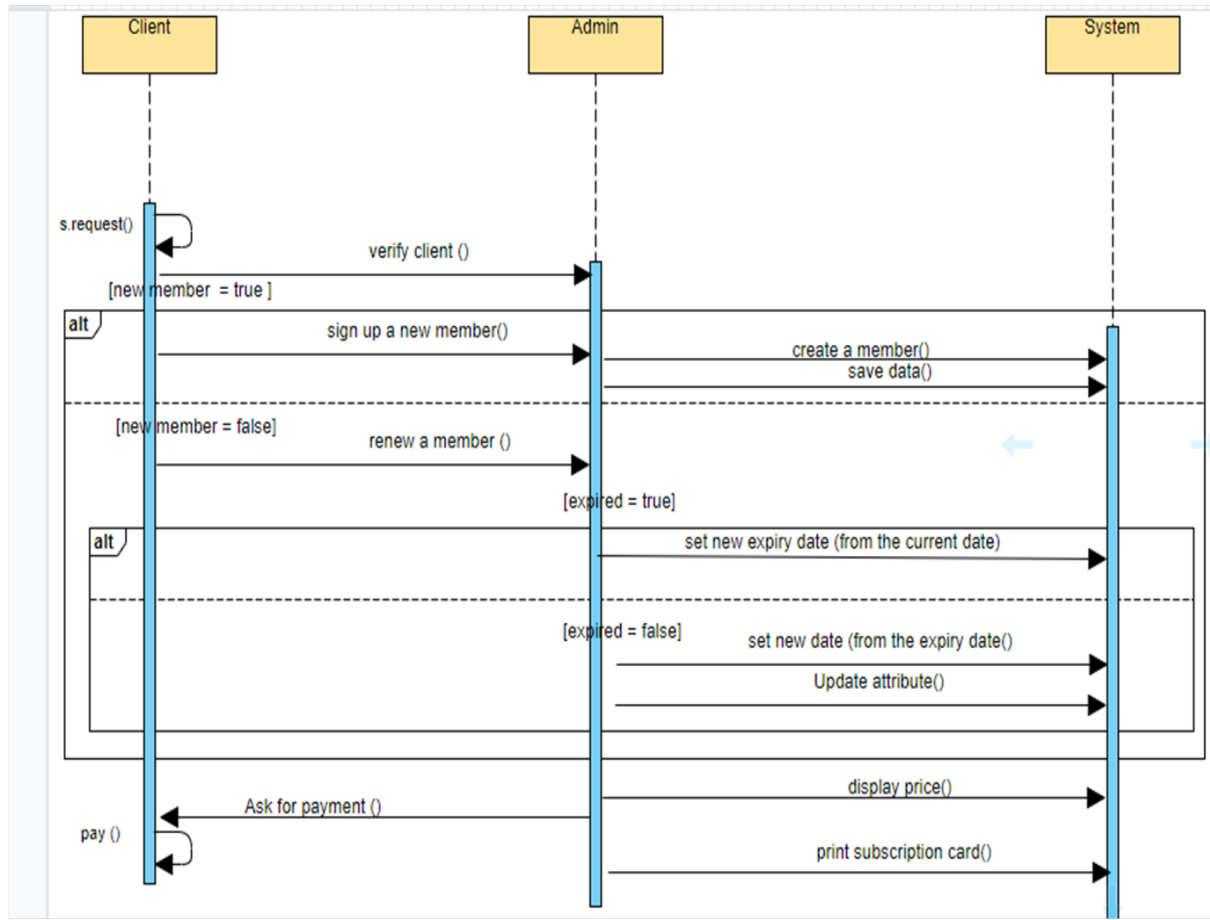
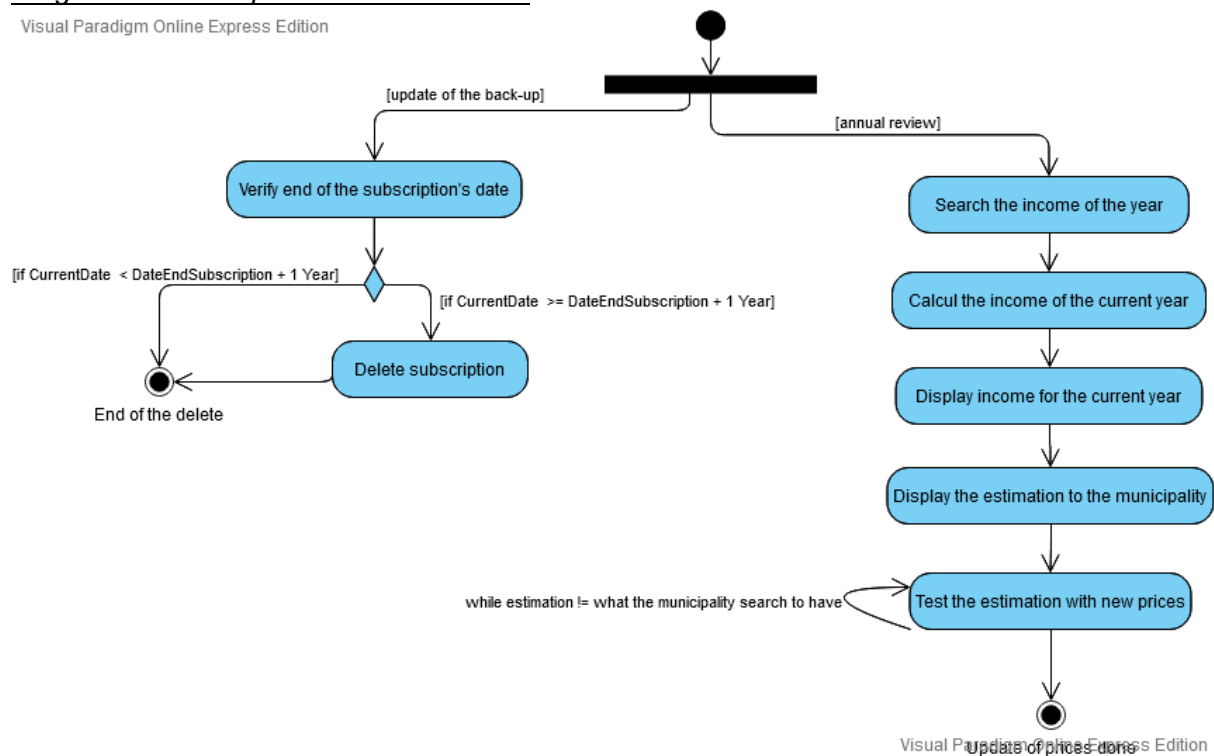


Diagramme de séquence : Abonnement

Visual Paradigm Online Express Edition



Visual Paradigm Online Express Edition

Diagramme d'activité : Autres requêtes de gestion

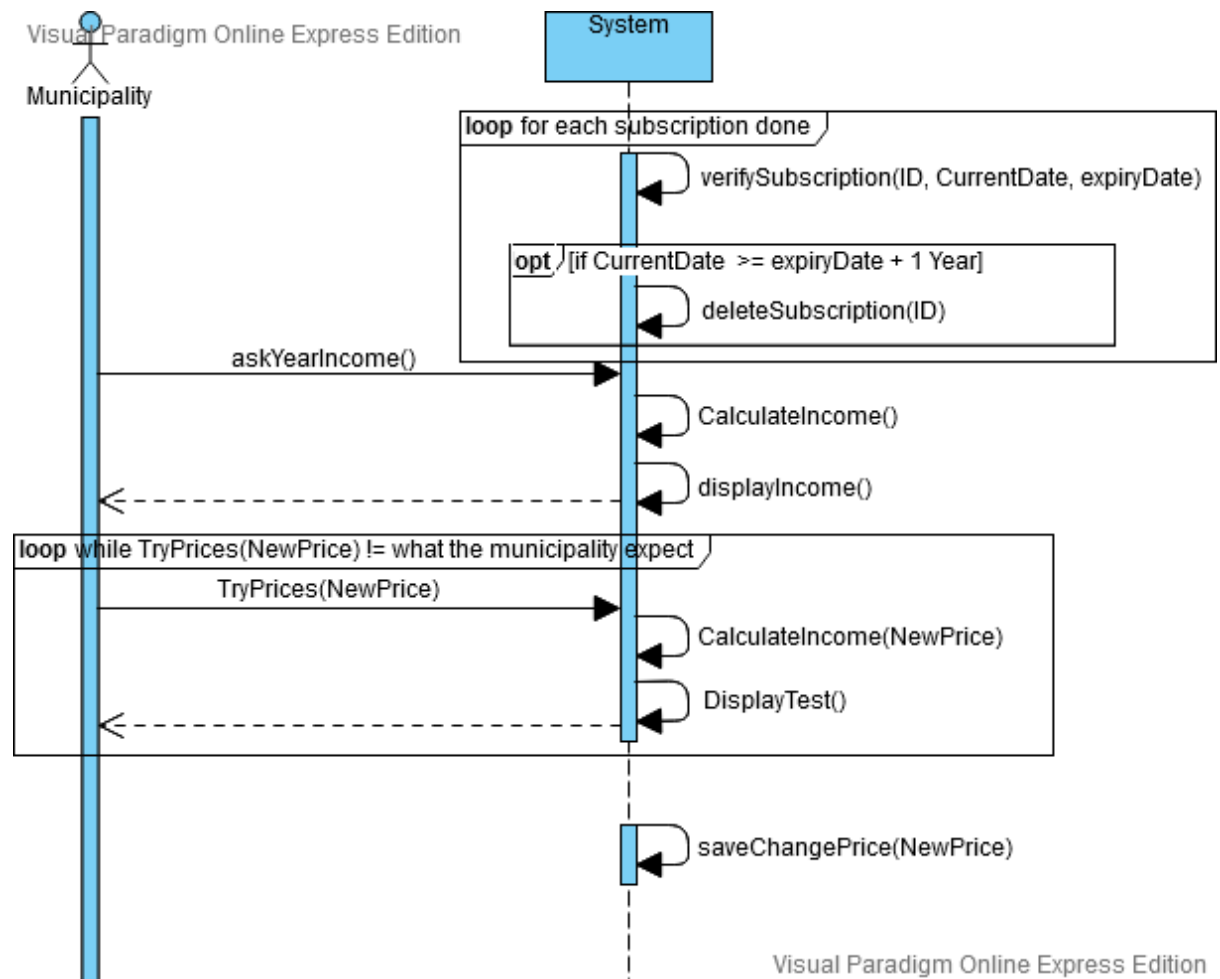
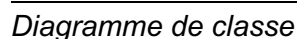
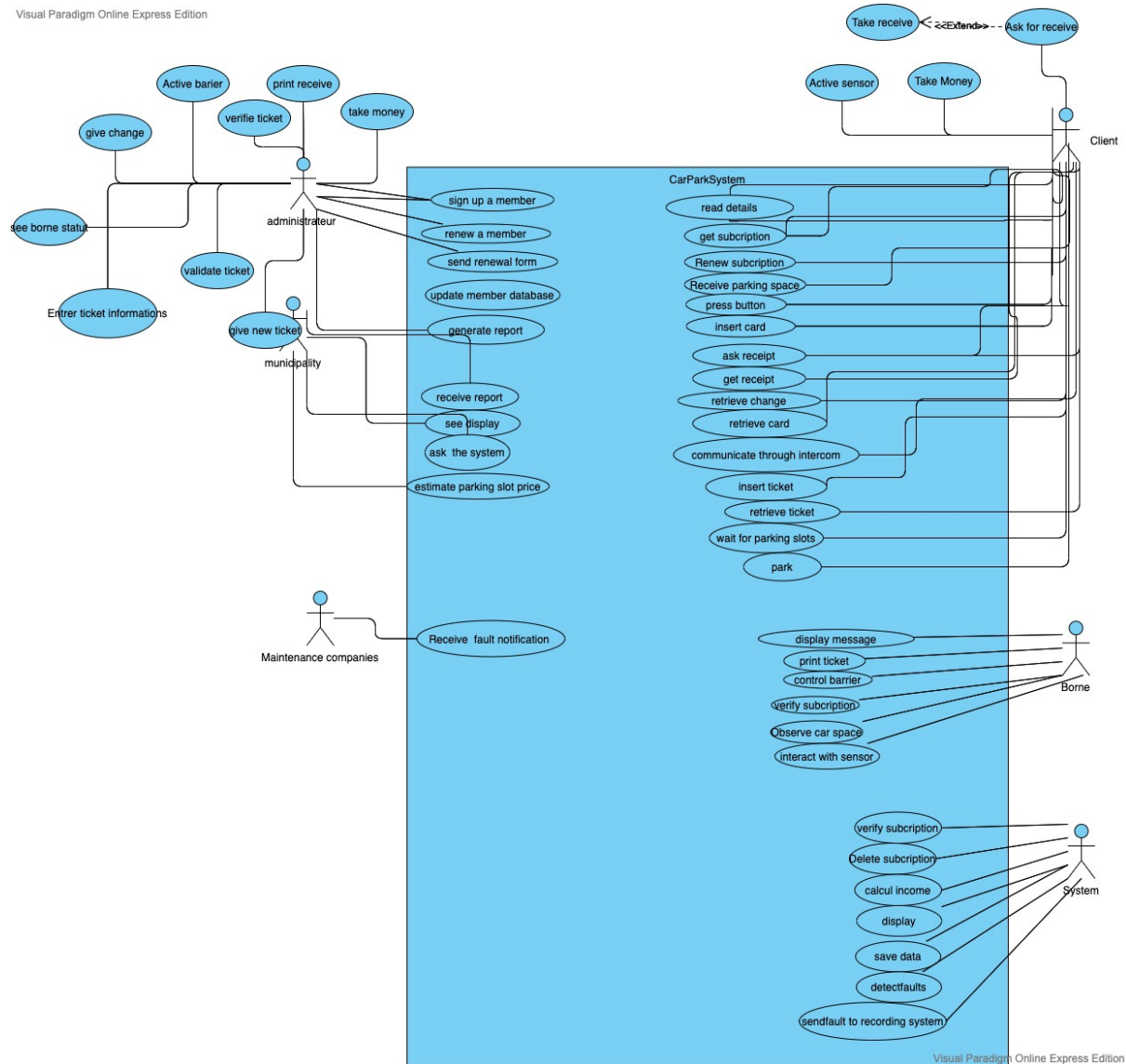


Diagramme de séquence : Autres requêtes de gestion





Use-case

Description textuelles des cas d'utilisations

Sommaire d'identification

Titre: Système opérationnel

Résumé: Gère l'entrée, la sortie et le paiement du parking

Acteurs : Abonné, Non abonné, borne

Sommaire d'identification

Titre: Entrée dans le parking d'un abonné

Résumé: Gère les conditions d'entrée d'une personne ayant un abonnement parking.

Acteurs : Abonné, Borne

Responsable : Perle

Description des scénarios

Préconditions :

- Jours de semaine (lundi - vendredi)
- Parking de l'abonnement
- Abonnement non expiré
- Abonnement non en cours d'utilisation (possesseur enregistré comme rentré, mais non sortie)

Scénario nominal :

1. La voiture roule jusqu'au capteur.
2. La borne indique "Appuyer sur le bouton"
3. L'abonné introduit sa carte dans la fente de la borne de contrôle
4. La borne vérifie les préconditions
5. La borne rend la carte à l'abonné
6. La borne enregistre l'heure d'entrée du possesseur de l'abonnement
7. La barrière se lève
8. L'abonné entre dans le parking
9. Le capteur détecte quand la voiture est passée et la barrière s'abaisse
10. Le nombre de véhicules dans les places allouées est incrémenté de 1
11. Comparaison du nombre de voitures dans le parking avec la capacité du parking

Enchaînements alternatifs :

A1 : Si la vérification ne permet pas l'entrée

L'enchaînement de A1 commence au point 5.

6. La borne affiche le message d'erreur.
7. La borne procède comme pour un client ordinaire.

Enchaînements d'erreur :

E1 : Nous sommes un jour du week-end

L'enchaînement de E1 commence au point 5.

6. La borne n'enregistre pas l'échec
7. La borne indique au client qu'on est le week-end

E2 : Abonnement pour un autre parking

L'enchaînement E2 commence au point 5

6. La borne n'enregistre pas l'échec
7. La borne indique au client qu'il n'est pas abonné à ce parking

E3 : Abonnement expiré

L'enchaînement E3 commence au point 5

6. La borne n'enregistre pas l'échec.

7. La borne indique au client que son abonnement n'est plus valide.

E4 : Abonnement en cours d'utilisation

L'enchaînement E4 commence au point 5.

6. La borne n'enregistre pas l'échec.

7. La borne indique au client que son abonnement est en cours d'utilisation

Postconditions :

Le nombre de place alloué pour les abonnés diminue, mais puisqu'ils ont chacun leur place attribués, ce n'est pas réellement à prendre en compte. Surtout, car les abonnements ne fonctionnent qu'en semaine.

Exigences non fonctionnelles

Contraintes - Descriptions

Contraintes	Descriptions
Temps de réponse	La borne doit faire les vérifications en moins de 5 secondes
Disponibilité	Le parking est accessible avec abonnement que les jours de semaine
Intégrité	L'entrée du parking ne doit pas pouvoir être forcée.

Besoins interface homme machine

Les dispositifs d'entrée et sortie à la disposition du client doivent être :

- Un lecteur de carte d'abonnement
- Un bouton (en cas d'erreur)
- Un écran pour l'affichage des messages de la borne
- Une barrière
- Un capteur

Sommaire d'identification

Titre: Entrée dans le parking d'un client ordinaire

Résumé: Gère les conditions d'entrée d'une personne n'ayant pas d'abonnement parking.

Acteurs : Client ordinaire, Borne

Responsable : Perle

Description des scénarios

Préconditions :

- Place libre dans les 90% non alloués aux abonnés en semaine et parmi toutes les places le week-end
- Papier d'impression des tickets

Scénario nominal :

1. La voiture roule jusqu'au capteur.
2. La borne affiche "Appuyez sur le bouton"
3. Le client appuie sur le bouton
4. La borne imprime un ticket et affiche "Prenez le ticket"
5. Le client prend le ticket
6. La barrière se lève
7. Le client entre dans le parking
8. Le capteur détecte quand la voiture est passée et la barrière s'abaisse

9. Le nombre de véhicules dans les places disponibles est incrémenté de 1
10. Comparaison du nombre de voitures dans le parking avec la capacité du parking

Enchaînements alternatifs :

A1 : Si le parking est complet

L'enchaînement de A1 commence au point 3.

4. La borne ne distribue pas de ticket
5. La borne affiche le message "Complet".
6. La barrière ne se lève pas.
7. La borne affiche de nouveau "Appuyez sur le bouton" que lorsqu'il y a de la place qui se libère.

Enchaînements d'erreur :

Il n'y a pas d'enchaînement d'erreur

Postconditions :

Après l'incrémentation du nombre de voiture dans le parking, on compare le nombre de voitures au nombre de places disponibles pour savoir si le parking est complet ou non.

Exigences non fonctionnelles

Contraintes - Descriptions

Contraintes	Descriptions
Temps de réponse	La borne doit imprimer les tickets en moins de 5 secondes
Disponibilité	Le parking est accessible tous les jours. Il ne doit pas manquer de papier d'impression des tickets de parkings
Intégrité	L'entrée du parking ne doit pas pouvoir être forcée.

Besoins interface homme machine

Les dispositifs d'entrée et sortie à la disposition du client doivent être :

- Un bouton
- Un distributeur de tickets
- Un écran pour l'affichage des messages de la borne
- Une barrière
- Un senseur

Sommaire d'identification

Titre: Sortie de parking d'un client ordinaire

Résumé: Gère la sortie du parking d'un client ordinaire

Acteurs : Client non abonné, borne, employé du bureau

Responsable : Nour, (Perle)

Description des scénarios

Préconditions :

- Être dans le parking
- Avoir un ticket
- Avoir payé le ticket il y a moins de 15 minutes

Scénario nominal :

1. La voiture roule jusqu'au senseur.

2. Le message "Insérer le ticket" s'affiche sur la borne.
3. Insertion du ticket dans la fente.
4. Lecture du code barre et vérification du paiement
5. La barrière se lève
6. Le capteur détecte le passage de la voiture
7. La barrière s'abaisse
8. Le nombre de véhicules dans le parking est décrétementé.
9. Comparaison de la capacité du parking avec le nombre de voitures

Enchaînements alternatifs :

A1: Si plus de 15 minutes se sont écoulés depuis l'heure

L'enchaînement A1 démarre au point 5

5. Activation de l'interphone dans la borne de contrôle
6. Connexion avec l'employé du bureau
7. L'employé active la barrière à distance

Le scénario reprend au point 5.

Enchaînements d'erreur :

Aucun enchaînement en cas d'erreur

Postconditions :

Après décrémentation du nombre de voiture dans le parking, on compare le nombre de voitures au nombre de places disponibles pour permettre l'accès au parking pour le prochain véhicule en attente.

Exigences non fonctionnelles

Contraintes - Descriptions :

Contraintes	Descriptions
Disponibilité	Il faut qu'il est constamment aux horaires du parking un employé au cas où les 15 minutes sont dépassées.
Intégrité	Il faut que la barrière soit baissée et qu'elle ne puisse pas être forcée.

Besoins interface homme machine :

Les dispositifs d'entrée et sortie à la disposition du client doivent être :

- Un lecteur du ticket
- Un interphone (en cas de problème)
- Une barrière
- Un capteur

Sommaire d'identification

Titre: Sortie de parking d'un abonné

Résumé: Gère la sortie du parking d'un abonné

Acteurs : Client abonné, borne, employé du bureau

Responsable : Nour, (Perle)

Description des scénarios

Préconditions :

- Être dans le parking
- Être abonné du parking

Scénario nominal :

1. La voiture roule jusqu'au capteur.
2. Le message "Insérer le ticket" s'affiche sur la borne.
3. Insertion du ticket d'abonnement dans la fente.
4. La barrière se lève
5. Enregistrement de l'heure de sortie de l'abonné
6. Le capteur détecte le passage de la voiture
7. La barrière s'abaisse
8. Le nombre de véhicules dans le parking est décrétementé.
9. Comparaison de la capacité du parking avec le nombre de voitures

Enchaînements alternatifs :

Aucun enchaînement alternatif

Enchaînements d'erreur :

Aucun enchaînement en cas d'erreur

Postconditions :

Décrémentation du nombre de voiture dans le parking pour les places allouées, ce qui n'affecte pas le nombre de place total du parking, puisque le nombre de carte d'abonnement correspond au nombre de place alloués. Et puisque les abonnements ne fonctionnent qu'en semaine.

Exigences non fonctionnelles

Contraintes - Descriptions :

Contraintes	Descriptions
Disponibilité	Il faut qu'il est constamment aux horaires du parking un employé au cas où les 15 minutes sont dépassées.
Intégrité	Il faut que la barrière soit baissée et qu'elle ne puisse pas être forcée.

Besoins interface homme machine :

Les dispositifs d'entrée et sortie à la disposition du client doivent être :

- Un lecteur du ticket
- Un interphone (en cas de problème)
- Une barrière
- Un capteur

Sommaire d'identification

Titre: Enregistrement de visite de sécurité

Résumé: Le municipalite a un contrat avec les entreprises de surveillance, d'après ce contrat,des gardes doivent être envoyer par l'entreprise au parking.Cette visite est enregistrée.

Acteurs : Garde, bornes de contrôle,lecture de carte

Responsable : Nozithelo

Description des scénarios

Scénario nominal :

1. La municipalité signe le contrat avec une entreprise de surveillance.
2. L'entreprise envoie un garde pour visiter le parking
3. Le garde arrive dans le parking
4. Il entre (en utilisant une carte donnée par son entreprise) et se gare s'il est en voiture.

5. Il entre dans le bureau ou se trouve la lecture de carte
6. Il insère sa carte dans la lecture de carte
7. La date et l'heure de sa visite sont enregistrés
8. Il quitte après une certaine durée, spécifiée dans le contrat.

Enchaînements alternatifs :

Si le garde est à pied, l'étape 4 ne sera pas effectuée.

Enchaînements d'erreur :

Aucun enchaînement en cas d'erreur

Postconditions :

Il n'y a pas de postcondition

Exigences non fonctionnelles

Contraintes - Descriptions :

Contraintes	Descriptions
Disponibilité	Le garde doit être disponible le temps minimum d'une visite, comme indiqué dans le contrat

Besoins interface homme machine

Les dispositifs d'entrée et sortie à la disposition du client doivent être :

- lecture de carte
- Borne
- Carte

Sommaire d'identification

Titre: Enregistrement des fautes

Résumé: Le système du parking englobe tout ce qui est interne au parking à savoir le système d'enregistrement de fautes qui est l'objet de cette partie. En effet, il a pour fonction principale de vérifier chaque appareil du parking tout détectant les fautes qui peuvent être occasionnées par exemple par les entrées et sorties des utilisateurs.

Acteurs : Entreprises de maintenance

Responsable : Amy

Description des scénarios

Préconditions :

Compte tenu du fait que le système d'enregistrement des fautes est interne au parking, ledit système ne pourra enregistrer les fautes que si le système du parking est fonctionnel. En d'autres termes, pour qu'il y ait enregistrement de fautes, il faudrait que le système du parking soit commode pour être à mesure d'envoyer des notification en fonction de la nature des fautes aux entreprises adéquates. Nous tenons à souligner que toutes les fautes engendrées ne peuvent pas empêcher l'utilisateur d'accéder au parking hormis l'expiration de son abonnement et le bug au niveau de tout le système.

Scénario nominal :

1. L'utilisateur arrive et insère le ticket dans la borne, si ticket valide, impression du ticket et levée de barrière (partie susmentionnée mais essentielle pour introduire cette sous-partie).
2. Le système du parking assure le fonctionnement de tous les systèmes internes (borne, système d'enregistrement de fautes, la barrière y compris le personnel) détecte les fautes internes au parking.

3. Si détection de fautes, les fautes sont envoyées au système d'enregistrement de fautes qui les enregistre y compris la date et les heures auxquelles, elles ont eu lieu.
4. Ensuite des notifications aux entreprises (équipe de maintenance ou groupe de maintenance de la municipalité).
5. Réparation des fautes par l'entreprise de maintenance adéquate.
6. Et enfin, production d'un rapport mensuel de toutes les fautes .

Enchaînements alternatifs :

Dans le cas où tout le système est non fonctionnel, aucune faute ne pourra être enregistrée, aussi aucune barrière ne pourra être levée et enfin aucune notification ne pourra être envoyée aux services de maintenance.

Enchaînements d'erreur :

Nous ne pensons pas qu'il pourrait en avoir.

Postconditions :

Tout le système du parking doit être fonctionnel.

Exigences non fonctionnelles

Contraintes - Descriptions :

Contraintes	Descriptions
Temps de réponse	De manière automatique
Disponibilité	Le système du parking doit être fonctionnel

Sommaire d'identification

Titre: Accord de niveau de service de surveillance

Résumé : Cette partie traite les rapports existants entre les entreprises de maintenance et le système du parking dans le cas des fautes causées par les entrées et sorties des utilisateurs. Un rapport de failles de toutes les occurrences du nombre de visites journalières du parking doit être produit par le système.

Acteurs : Entreprises de maintenance

Responsable : Amy

Description des scénarios

Préconditions :

- Enregistrement des entrées du parking dans le système.

Scénario nominal :

Cette sous-partie est subtilement rattachée à la sous-partie "Enregistrement des fautes".

1. Production d'un rapport de failles(fautes) causées par toutes les occurrences du nombre de visites journalières du parking.

Enchaînements alternatifs :

Les occurrences du nombre de visites journalières sont nécessaires pour faire les rapports des failles (fautes).

Enchaînements d'erreur :

Nous n'en trouvons pas dans cette sous-partie.

Postconditions :

Nous n'en trouvons pas dans cette sous-partie.

Sommaire d'identification

Titre: Abonnements

Résumé: Un sous-système qui gère la vente des abonnements.

Acteurs : Un client, le système et un administrateur

Responsable : Nozithelo

Description des scénarios

Préconditions :

Il n'y a pas de précondition spécifique pour la création d'un abonnement, cependant pour le renouvellement d'un abonnement, un formulaire doit être envoyé deux semaines avant la date d'expiration.

Scénario nominal :

1. Requête d'abonnement
2. Si le client est nouveau l'administrateur crée un nouveau membre
3. Le système affiche l'interface et créer un nouveau membre
4. Admin renseigne des information sur le client
5. Le système sauvegarde les informations
6. Spécification d'un abonnement demandé (3, 6 ou 12 mois)
7. Le système calcule et affiche le prix de l'abonnement
8. Paiement de l'abonnement
9. L'enregistrement du paiement
10. Système générer et impression de la carte d'abonnement
11. Le client reçoit sa carte d'abonnement.

Enchaînements alternatifs :

A1: S'il ne s'agit pas d'un nouveau client

L'enchaînement A1 démarre au point 1

1. L'administrateur renouvelle l'abonnement
2. Si l'abonnement est déjà expiré, la date d'expiration est prolongée selon la durée d'abonnement choisi, à partir de la date de ce jour là (from the current date)
3. Si l'abonnement n'est pas encore expiré, la date est prolongée à partir de la date d'expiration, selon l'abonnement choisi.
4. Le système met à jour les informations (update attributes)
5. Les étapes 7,8 et 9 du scénario précédent sont effectuées.

Enchaînements d'erreur :

E1: Si le client ne met pas assez d'argent pour payer

L'enchaînement E1 démarre au point 8

9. L'abonnement est annulé.

Postconditions :

L'abonnement est créé et utilisable par l'abonné.

Exigences non fonctionnelles

Besoins interface homme machine :

Les dispositifs d'entrée et sortie à la disposition du client doivent être :

- une imprimante
- un ordinateur

Sommaire d'identification

Titre: Autres requêtes de gestion

Résumé: Gère les abonnements expirés depuis longtemps et permet de mettre à jour le prix des différents tarifs.

Acteurs : Municipalité, Système

Responsable : Perle

Description des scénarios

Préconditions : Il y a des abonnements

Scénario nominal :

1. Le système vérifie les dates de fin d'abonnement
2. Si la fin de l'abonnement date d'il y a plus d'un an, on le supprime
3. La municipalité regarde dans les données le niveau de revenu
4. Une estimation du niveau de revenu de l'année en cours est faite
5. Le système affiche cette estimation
6. La municipalité teste cette estimation avec leur nouveaux tarifs.
7. La municipalité change leur tarif.

Enchaînements alternatifs :

A1 : L'estimation avec leur nouveaux tarifs n'est pas comme souhaité

On répète le point 7 jusqu'au moment où les nouveaux tarifs correspondent au revenu souhaité par la municipalité pour l'année suivante.

Enchaînements d'erreur :

Il n'y a pas d'enchaînement en cas d'erreur

Postconditions :

Les tarifs du parking ont évolué.

Exigences non fonctionnelles :

Besoins interface homme machine :

Les dispositifs d'entrée et sortie à la disposition du client doivent être :

- Ordinateur contenant les données