Rapport  
  
Le plan du rapport :  
Intro : -rappel du sujet     OK  
Hypothese, reformulation: OK  
-cas utilisation OK  
-hierarchiser les fonctionnalités (ca veut dire quoi ?)  
Diagramme de classes: OK (on pourra meme parler de notre evolution grace a 3 diag de classe haha x) )  
Modelisation dynamique:  
-sequence OK  
-activite (en cours)  
- etat transition (on fait ?)

Réservation de billets d’avion

Ce projet vise à proposer une application de réservation de billets d’avion destinée à être une compagnie aérienne pour gérer les réservations des clients et optimiser ses coûts.

La compagnie aérienne dispose d’une flotte d’avions qui desservent différents aéroports. Aucun avion n’est affecté à une liaison particulière et tous peuvent effectuer presque tous les trajets. Chaque avion possède une capacité maximale de passagers et présente une autonomie maximale (exprimée en km par exemple). Nous attribuerons également des coûts à chaque vol : faire voler un gros avion peu rempli reviendra par exemple plus cher à la compagnie qu’utiliser un petit avion plein.

L’application doit ainsi permettre à la compagnie aérienne :

* trouver un trajet reliant deux aéroports;
* d’ajouter/supprimer des voyageurs sur un vol;
* de fournir un récapitulatif d’un voyage;
* de choisir l’avion le plus économique pour un vol donné.

Vous proposez enfin des visualisations graphique les destinations desservies (par jour, semaine ou mois, éventuellement en ajoutant la référence de l’avion effectuant la liaison).

1. Contexte et rappel du sujet

Dans un monde (contexte ?) grandissant de la démocratisation des échanges des hommes et des marchandises grâce notamment aux routes aériennes pour le tourisme ou l’économie, la mise en place d’un système de réservation efficace est essentielle pour facilités les besoins des clients et des compagnies.

Dans ce but, il nous a été demandé de faire une application de réservation de billets d’avion destinée à être utilisée par les compagnies aériennes pour gérer les réservations mais également par les clients pour qu’ils saisissent les critères de vol eux-mêmes et choisissent le vol qu’ils leur conviennent le mieux. De plus si possible il serait pertinent d’avoir une visualisation de quelques fonctionnalités comme les différents vols disponibles, un plan de vol .. Il s’agit donc de faire un produit s’apparentant à un CRS[[1]](#footnote-1) « computer reservations system » ou un GDS[[2]](#footnote-2) « Global distribution system » pour une utilisation du client ou de la compagnie aérienne.

1. Hypothèses et reformulation

Afin de comprendre aux mieux les problèmes possibles liés à une réservation nous avons listé les différents besoins des deux acteurs qui nous ont semblés important et ce qu’il serait intéressant de visualiser. Nous avons regroupé ce que nous avons retenu dans le tableau suivant :

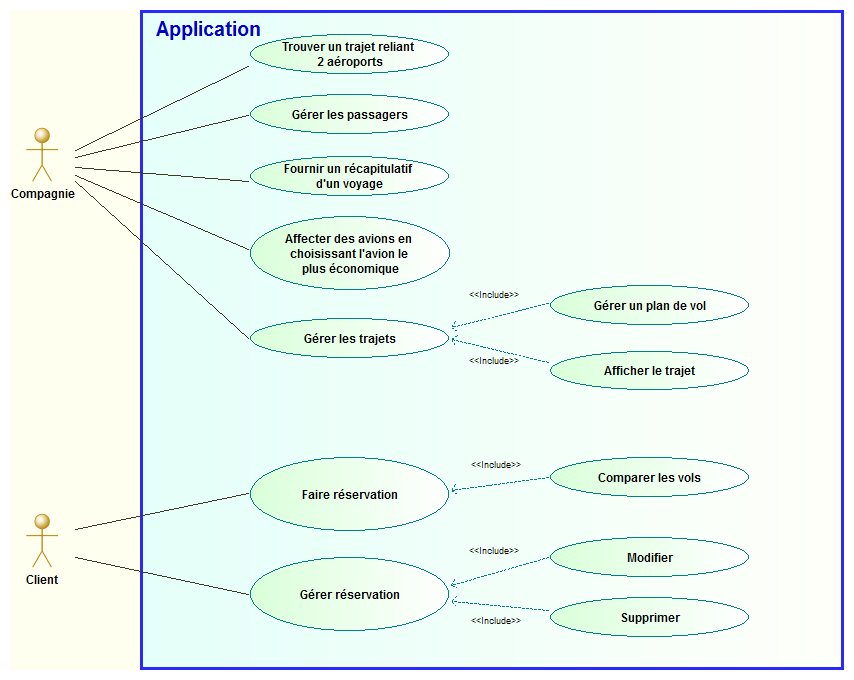
|  |  |  |
| --- | --- | --- |
| Client : Réservations de billets | Compagnie : Gérer la compagnie | Visualiser |
| - Proposer des choix (dates, destinations, sièges,..)  - Faire un billet  - Gérer sa réservation (modifier, annuler…) | - Affecter des avions (capacité, autonomie, coût : remplissage, consommation …)  - Gérer les routes (taux remplissage, taxes..)  - Gérer un plan de vol  - Gérer les passagers (affecter aux vols…)  (- Promos) | - Vols disponibles  (en fonction de destination, dates, promos..)  - Routes (sur la carte)  - Trajectoire des avions (plan de vol, position, en fonction du temps) |

Etant tous les deux assez familiers avec les réservations de billet d’avion grâce à nos voyages nous avons eu certaines idées pour améliorer les réservations et les rendre ainsi plus intéressantes avec une fidélisation des clients. Ces fonctionnalités pourraient être mises en place si nous avons assez de temps :

* fidéliser son client, en lui attribuant des points (comme des miles)
* faire des promos lors de période de creux (hors vacances par exemple
* proposer des hôtels avec les destinations choisies pour faciliter les recherches du client et une voiture si besoin
* proposer une assurance en cas d’imprévus
* proposer de la flexibilité au niveau des dates et d’autres options mais cela pourrait engendrer des frais supplémentaires (repas particulier, ..)
* il pourrait être astucieux de montrer le nombre de points que la réservation permet de gagner et de montrer l’itinéraire aller-retour

Diagramme de cas d’utilisation

Nous avons ainsi réalisé le diagramme d’activité suivant en intégrant seulement les fonctionnalités de base que devrait faire l’application:



Les deux acteurs principaux ont chacun des actions réalisables distinctes. Le client s’occupe exclusivement de sa réservation : une fois la réservation faite, il peut la modifier ou la supprimer. Tandis que la compagnie peut effectuer plusieurs actions liées aux vols, aux avions, aux routes .. La compagnie peut trouver un trajet reliant 2 aéroport, gérer les passagers, fournir un récapitulatif de voyage avec toutes les informations concernant sa destination, son vol etc.. La compagnie peut affecter des avions en choisissant l’avion le plus économique et gérer les trajets en générant un plan de vol et l’affichant.

Les hypothèses

Au moment-même de la rédaction de notre rapport d’analyse toutes nos hypothèses ne sont pas encore définitives, certaines nécessitent une réflexion plus poussée en corrélation avec une programmation pour se rendre réellement compte des hypothèses nécessaires, en voici une partie non-exhaustives : (c’est un peu too much non cette formulation, je trouve l’utilise pas bien ?)

- Ainsi pour l’instant la gestion des avions au tout début du programme n’est pas encore bien définie, nous supposons donc que dans un premier temps lors du lancement de l’application, il y a assez d’avions, là où il faut pour les réservations.

- De plus nous supposons que le client débutant une réservation va nécessairement la mener à son terme, c’est-à-dire sans s’arrêter en plein milieu de la réservation.

- Les avions se déplacent le long d’une ligne géodésique et à vitesse constante.

- Nous travaillons avec une base de données où nous avons seulement gardé les grands et moyens aéroports.

- Nous avons les horaires que de certaines compagnies aériennes, nous les regroupons donc en grande banque d’horaire où les horaires sont identiques au sein d’une même banque.

- Le prix du billet sera fixer en parti par les aéroports : plus l’aéroport a de nombre de routes qui lui sont connectés, plus il est intéressant donc le prix sera plus élevé, les lowcost utilise par exemple des aéroports moins grand puisqu’ils seront moins cher.

- On hésite encore sur le « temps » de l’application, est-ce qu’on applique une horloge interne différente à l’horloge absolue de la Terre, est-ce que tout revient à 0 ou est-ce que on le fait marcher l’application en continu. Nous penchons plus vers l’idée d’une horloge interne (c’est-à-dire une échelle de temps propre au programme) à l’application, elle serait la meilleure solution en attendant de trouver une autre façon de penser.

- Nous nous sommes limités à la création de 23 types d’avions.

La hiérarchie des fonctionnalités

|  |  |  |
| --- | --- | --- |
| Les fonctionnalités essentielles que permet l’application sont : | | |
| - pour a compagnie de : | - trouver un trajet reliant 2 aéroports  - gérer les passagers  - fournir un récapitulatif d’un voyage  - affecter des avions en choisissant l’avion le plus économique | |
| - pour le client de : | - faire une réservation  - gérer sa réservation | |
| Les fonctionalités secondaires sont : | | |
| - pour les compagnies de : | | - gérer les trajet : gérer un plan de vol, afficher le trajet |
| Les fonctionnalités facultatives que l’on aimerait apportées sont de : | | |
|  | | - fidéliser son client, en lui attribuant des points (comme des miles)  - faire des promos lors de période de creux (hors vacances par exemple  - proposer des hôtels avec les destinations choisies pour faciliter les recherches du client et une voiture si besoin  - proposer une assurance en cas d’imprévus  - proposer de la flexibilité au niveau des dates et d’autres options mais cela pourrait engendrer des frais supplémentaires (repas particulier, ..)  - il pourrait être astucieux de montrer le nombre de points que la réservation permet de gagner et de montrer l’itinéraire aller-retour |
|  | |  |

-

1. Diagramme de classe et modélisation

|  |
| --- |
| E:\Projet informatique\Projet resavion\Projet-Info.git\Diagramme de classe version 1.png  Diagramme de classe version 1 |
| C:\Users\Flora\Downloads\Diagramme de classe version 2.png  Diagramme de classe version 2 |

|  |
| --- |
| C:\Users\Flora\Downloads\Diagramme de classe version 3.png  Diagramme de classe version 3 (dernière version) |

Au fur et a mesure de notre réflexion et de notre modélisation nous avons changé petit a petit notre diagramme de classe. On peut ainsi voir 3 grandes phases evolutives avec la dernière qui est à terme celle que nous avons gardé. Cependant avec l’avancé de notre programmation elle ne correspondra pas exactement à notre code puisque l’on s’est rendu compte que quelques ajouts étaient encore nécessaires.

1. Modélisation dynamique

Pour comprendre comment nous allions l’enchainement logique d’une réservation faite par un client nous avons fait plusieurs simulations de réservations ce qui a abouti à la stratégie suivante :

* Demander le lieu (aéroport) de départ, demander le lieu (aéroport) d’arrivée
* Demander une date de départ, une date de retour
* Demander le nombre de passagers (adulte/enfant)
* Demander quel type de classe (eco, eco plus, business, première)

-¤ heure de départ/ heure de retour

-¤ nombre d’escales

-¤ compagnie voulue

* Affichage des différents vols possibles avec les différents horaires
* Remplissage des informations des passagers une fois le choix effectué

-¤ choisir le siège dans l’avion avec affichage du plan

-¤ rentrer un code promo

-¤ accepter les conditions

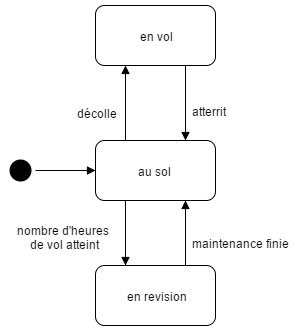
-¤ proposer une assurance de voyage/ assurance de flexibilité

* Demander si le passager à besoin de nécessité particulière (voyager avec animal, mobilité réduite, mineur non accompagné, autre demande ..)
* Demander le payement
* Envoie du compte-rendu de la réservation

Les étapes ayant « -¤ » devant ne sont pas des étapes essentielles à la réservation mais des options qui nous sont venus en tête pour affiner la réservation et correspondre au mieux aux besoins du client et qui seront des fonctionnalités avancées.

|  |
| --- |
| Diagramme de séquence d’une réservationC:\Users\Flora\Downloads\Diagramme de séquence reservation (1).png |

|  |
| --- |
| C:\Users\Flora\Downloads\Diagramme d'activite d'une reservation.png |



1. CRS : ils permettent de réserver un service dans le temps pour une personne ou un groupe. Ces systèmes s'intègrent dans plusieurs domaines notamment celui du transport tel que l’avion. Ils peuvent être directement accessibles aux personnes non professionnelles grâce au développement de serveurs web accessibles depuis un navigateur. [↑](#footnote-ref-1)
2. GDS : ces plates-formes électroniques de gestion des réservations qui permettent aux agences de voyages de connaître en temps réel l'état du stock des différents fournisseurs de produits touristiques comme les compagnies aériennes et de réserver à distance. [↑](#footnote-ref-2)