Université de Mons
Faculté Des Sciences
Département d'Informatique

Projet de modélisation logicielle

Description semi-structurée des use-cases

Professeur :
Tom Mens
Assistants :
Jeremy Dubrulle
Gauvin Devillez
Sébastien Bonte

Auteurs :
Pignozi Agbenda
Thomas Bernard
Théo Godin
Ugo Proietti

Année académique 2021-2022

Table des matières

1 Applications de base

1.1 Use-cases communs aux 2 applications

1.2 Use-cases Application 1

1.3 Use-cases Applications 2

2 Extension 1 : Bernard Thomas

2.1 Use-cases de l'application 1

2.1.1 Demander un devis pour un type d'assurance

Acteurs : Le client, l'institution et le serveur

Descrition : Permet au client de demander un devis pour un certain type d'assurance ou plusieurs types d'assurance.

Préconditions : Le client se trouve sur l'écran lié à la demande de devis.

Fréquence : De temps en temps (lorsque le client désire un devis).

Parcours de base :

1. Le client clique sur l'option de demande d'un devis pour une assurance.
2. Le client choisit l'assurance propre à l'institution pour laquelle il désire avoir un devis.
3. L'application envoie les informations au serveur.
4. Le serveur renvoie le devis.
5. Le client consulte le devis.

Postconditions : La GUI affiche le devis que le client a demandé.

2.1.2 Obtenir des informations sur les différents types d'assurances

Acteurs : Le client et le serveur

Description : Permet au client d'obtenir des informations sur toutes les assurances que propose l'insitution financière à laquelle est lié le portefeuille sélectionné.

Préconditions : Le client doit être connecté, avoir sélectionné un de ses portefeuilles être entré dans la gestion des produits financiers.

Fréquence : De temps en temps.

Parcours de base :

1. Le client clique sur la fenêtre liée aux informations concernant les assurances.
2. L'application envoie la demande d'informations au serveur avec l'institution pour laquelle elles sont demandées.
3. Le serveur renvoie la liste des assurances de l'institution ainsi que tous les détails les concernants.
4. Le client consulte les informations.

Postconditions : La GUI affiche la liste des assurances pour l'institution sélectionnée ainsi que toutes les informations qui en découlent.

2.1.3 Accéder à la liste des assurances :

Acteurs : Le client et le serveur.

Description : Un client accède à la liste des assurances auxquelles il a soucrit dans l'institution qui correspond au portefeuille sélectionné.

Préconditions : Le client doit avoir choisi un de ses portefeuille auquel il voulait accéder et doit avoir choisi de gérer ses produits financiers.

Fréquence : Assez souvent

Parcours de base :

1. Le client clique sur l'accès à ses assurances.
2. L'application envoie au serveur une demande de récupération des assurances pour le client en particulier.
3. Le serveur renvoie les diverses assurances auxquelles le client a soucrit s'il en a. Si pas le serveur ne renvoie rien.
4. Le client consulte ses assurances.

Postconditions : La GUI affiche la liste des assurances du client et la possibilité d'en rajouter.

Points d'extensions :

1. Souscrire à une assurance

Parcours alternatif : Le client coche la case permettant d'afficher également les assurances qu'il a résilié. Et le serveur renvoie également les assurances qui sont désactivées afin de les afficher.

2.1.4 Gérer une assurance :

Acteurs : Le client et le serveur.

Description : Le client accède à la gestion d'une des assurances auxquelles il a souscrit.

Préconditions : Le client doit avoir sélectionné une des assurances présentes dans sa liste d'assurances.

Fréquence : Peu souvent

Parcours de base :

1. Le client sélectionne une de ses assurances.
2. L'application envoie une demande au serveur pour recevoir les informations liées à cette assurance.
3. Le serveur renvoie les informations de cette assurance.
4. Le client se trouve dans la gestion de son assurance.

Postconditions : La GUI affiche l'assurance sélectionnée ainsi que toutes les options qui sont liées à celle-ci.

Points d'extension :

1. Résiliez l'assurance.
2. Verser/retirer de l'argent d'une assurance.

2.1.5 Visualiser l'historique :

Acteurs : Le client et le serveur

Description : Le client accède à l'historique concernant ses assurances. Il peut y voir ses paiements et ses retraits.

Préconditions : Le client doit se trouver dans la liste des assurances.

Fréquence : Peu souvent

Parcours de base :

1. Le client clique sur le bouton d'affichage de l'historique dans la fenêtre d'une assurance.
2. L'application envoie la demande d'historique spécifique au client au serveur.
3. Le serveur renvoie les informations liées à l'historique.
4. L'application affiche l'historique de l'assurance du client
5. Le client consulte son historique

Postconditions : La GUI affiche l'historique de l'assurance du client.

2.1.6 Souscrire à une assurance :

Acteurs : Le cleint et le serveur.

Description : Le client décide depuis la liste de ses assurances de sopuscrire à une nouvelle assurance dans l'insitution à laquelle sont portefeuille est lié.

Préconditions : Le client doit posséder un portefeuille dans l'institution en question et se trouver dans la liste de ses assurances.

Fréquence : Quelques fois (lorsque le client n'a pas encore d'assurances) et peu souvent lorsqu'il en a déjà.

Parcours de base :

1. Le client clique sur le bouton d'ajouter d'une assurance.
2. Le client est rediriger dans la fenêtre de souscription à une assurance.
3. Le client est amené à choisir l'assurance à laquelle il souhaite souscrire.
4. Le client sélectionne le compte avec lequel il souhaite effectuer le paiement de la prime.
5. L'application envoie les informations de la souscription et du paiement au serveur.
6. Le serveur vérifie que le client possède suffisamment de liquidités sur le compte sélectionné.
7. Le serveur effectue le débit du compte et met à jour la prochaine date de paiement de l'assurance.
8. Le serveur ajoute l'inscription à la liste des assurances du client en question.
9. Le serveur envoie la confirmation à l'application ainsi que les informations liées à l'assurance.
10. L'application renvoie le client sur la scène de la liste des assurances et affiche la nouvelle assurance dans la liste.

Postconditions : La GUI affiche la nouvelle assurance dans la liste des paiements ainsi que la prochaine date de paiement. La GUI met également à jour l'affichage du compte qui a été débité dans la liste des comptes du client.

Parcours alternatif : A partir de la vérification serveur.

1. Le serveur détecte qu'il n'y a pas assez d'argent sur le compte sélectionné par le client.
2. Le serveur annule l'ajout de l'assurance.
3. Le serveur renvoie un message d'erreur à l'application stipulant que le solde du compte est insuffisant.

4. L'application affiche l'erreur et demande au client de sélectionner un nouveau compte ou d'annuler se requête.
5. retour au parcours de base où le client choisit un compte.

2.1.7 Résiliez une assurance :

Acteurs : Le client et le serveur.

Description : Le client souhaite résiliez une assurance à laquelle il avait souscrit.

Préconditions : Le client doit est à au minimum 3 mois de la prochaine date de paiement de l'assurance sans quoi il devra payer la prime de celle-ci avant de pouvoir la résiliez.

Fréquence : Peu souvent.

Parcours de base :

1. Le client sélectionne l'assurance et dans le menu de celle-ci clique sur le bouton de suppression.
2. L'application demande une confirmation au client.
3. Le client effectue son choix, s'il annule il a ramené sur la visualisation de son assurance
4. L'application envoie au serveur la requête de suppression de l'assurance en question.
5. Le serveur contrôle la date de paiement de l'assurance.
6. Le serveur envoie la confirmation que la date est conforme aux critères de résiliation et désactive l'assurance dans la base de donnée.
7. Le serveur envoie la confirmation de suppression à l'application.
8. L'application renvoie le client sur la liste des assurances et lui confirme la résiliation.
9. Le client peut à présent consulter ses assurances et voir que son assurance est bien désactivée.

Postconditions : La GUI affiche bien que l'assurance est désactivée si le client affiche les assurance désactivées et n'affiche plus son assurance sans cette option.

Parcours alternatif :

1. Le serveur contrôle le temps restant avant le paiement restant de l'assurance.
2. Le serveur envoie une erreur à l'application stipulant que le client doit payer la prime de cette dernière car il est trop tard pour la résilier.
3. L'application affiche le message d'erreur et renvoie l'utilisateur sur le menu de l'assurance.

2.1.8 Payer la prime :

Acteurs : Le client et le serveur.

Description : Le client effectuer le paiement de sa prime que celui-ci soit de manière automatique ou bien manuelle.

Préconditions : Il doit se trouver dans le menu de l'assurance en question.

Fréquence : A chaque renouvèlement de l'assurance.

Parcours de base :

1. Le client clique sur la bouton pay dans le menu de l'assurance ou a sélectionné au préalable le paiement automatique.
2. Le client sélectionne le compte avec lequel il souhaite effectuer le paiement ou dans le cas d'un paiement automatique l'avait sélectionné lors de l'activation de la feature.
3. L'application envoie les informations de paiement au serveur.
4. Le serveur vérifie que le solde du compte sélectionné est suffisant.
5. Le serveur débite le compte, modifie la date de paiement en la mettant à une nouvelle échéance.
6. Le serveur envoie la confirmation à l'application avec les nouvelles informations.
7. L'application reçoit les informations met la date depaiement ainsi que le solde du compte à jour dans l'affichage.
8. Le client reçoit la confirmation du paiement.

Postconditions : La GUI affiche l'assurance comme payée et renouvèle la date de paiement. La GUI affiche le compte comme débité.

Parcours alternatif :

1. Le serveur vérifie le solde et celui-ci est insuffisant.
2. Le serveur envoie une erreur à l'application stipulant que le solde du compte sélectionné est insuffisant.
3. Si le client est connecté l'application lui affiche une erreur lui demande de sélectionner un autre compte de provisionner celui sélectionné. Si le client n'est pas connecté il sera immédiatement rediriger vers le paiement lors de sa prochaine connexion au portefeuil lié à l'assurance. Les mêmes options lui seront proposées.
4. On revient au point de vérification du solde dans le parcours de base.

2.1.9 Verser/retirer de l'argent d'une assurance :

Acteurs : Le client et le serveur.

Description : Dans le cas d'une assurance qui fonctionne avec des fonds déposés par le client(vie, pension). Le client peut décider d'ajouter de l'argent ou 'en retirer de son assurance.

Préconditions : Il doit s'agir d'une assurance qui fonctionne avec un principe de cottisation (Assurance vie, assurance pension, etc).

Fréquence : De temps en temps.

Parcours de base :

1. Le client se trouve dans la gestion de son assurance et décide d'ajouter ou bien de retirer des fonds en cliquant sur les boutons correspondant à ces actions.
2. L'application change de scène sur la scène de dépôt ou de retrait.
3. Dans le cas du dépôt le client choisit le montant ainsi que le compte qui doit être crédité. Dans le cas d'un retrait, le montant du retrait ainsi que le compte qui doit être provisionné.
4. L'application envoie les données récoltées au serveur.
5. Dans le cas d'un dépôt le serveur vérifie que le compte qui doit être crédité possède bien au minimum le montant. Dans le cas d'un retrait il vérifie que le montant de l'assurance est au minimum celui fournit.
6. Le serveur crédite/provisionne les comptes et assurances et met à jour les montants dans la base de données. Et ensuite renvoie les nouvelles informations à l'application ainsi qu'une confirmation.
7. L'application reçoit la confirmation et met à jour l'affichage en fonction des informations renvoyées.

Postconditions : La GUI affiche les modifications qui ont été effectuées et notifie l'utilisateur que l'opération a bien été effectuée.

Parcours alternatif :

1. Après vérification le serveur détecte qu'un solde n'est pas suffisant.
2. Le serveur envoie l'erreur à l'application et annule l'opération.
3. L'application affiche l'erreur à l'utilisateur et lui demande dans le cas d'un dépôt de slectionner un autre compte ou bien de l'approvisionner et dans le cas d'un retrait informe le client que le montant demandé n'est pas disponnible.
4. Le client sélectionne un autre compte/provisionne le compte ou annule l'opération.

2.2 Use-case Application 2

2.2.1 Répondre à une demande de devis

Acteurs : L'institution, le serveur, le client.

Description : L'institution génère un devis concernant un ou plusieurs types d'assurances afin de répondre à la demande du client.

Préconditions : Le client doit avoir effectué une demande de devis qui a été envoyée au serveur et que le serveur a envoyé à l'institution.

Fréquence : Rarement

Parcours de base :

1. L'insitution sélectionne l'onglet de demandes de devis faites par des clients.
2. L'insitution sélectionne une demande de devis en récupère les informations et lance la génération d'un devis avec les informations demandées par l'utilisateur.
3. L'application envoie la demande au serveur.
4. Le serveur analyses les informations dont il a besoin et retourne à l'applications toutes les assurances correspondant aux critères.
5. L'application affiche le devis généré à l'institution.
6. L'institution envoie le devis au client l'ayant demandé.
7. L'application envoie le devis au serveur.
8. Le serveur envoie le devis à l'application client.
9. L'application client traite la demande en l'envoyant au client.

Postconditions : Dans l'application instituion la demande de devis apparaît comme traitée au niveau de la GUI et dans l'application client le client recevra une réponse à celle-ci.

3 Extension 1.6.1 : gestion des cartes - Godin Théo

3.1 Use-case Application 1

3.1.1 Se connecter via une carte

Acteurs : Le client et le serveur

Description : Le client s'autentifie au moyen d'une de ses cartes de paiement.

Préconditions : Le client possède au moins une carte liée à une instituion.

Fréquence :rarement

Parcours de base :

1. L'application génère un code unique.
2. Le client insère sa carte dans le lecteur de carte.
3. Le client clique sur le bouton login du lecteur de carte et entre le code généré par l'application.
4. Le client entre son code pin.
5. Le lecteur de carte génère un code d'autentification.
6. Le client entre le code d'autentification dans l'application.

Parcours alternatif :