

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 :

Pignozzi AGBENDA

Thomas BERNARD

Théo GODIN

Ugo PROIETTI



Année académique 2021-2022

Table des matières

1	Applications de base	1
1.1	Use-cases communs aux 2 applications	1
1.2	Use-cases Application 1	1
1.3	Use-cases Applications 2	1
2	Extension 1 : Bernard Thomas	2
2.1	Use-cases de l'application 1	2
2.1.1	Demander un devis pour un type d'assurance	2
2.1.2	Obtenir des informations sur les différents types d'assurances	3
2.1.3	Accéder à la liste des assurances :	4
2.1.4	Gérer les paramètres :	5
2.1.5	Visualiser l'historique :	6
2.1.6	Souscrire à une assurance :	7
2.1.7	Résilier une assurance :	9
2.1.8	Effectuer une transaction :	10
2.1.9	Payer la prime :	11
2.1.10	Verser/retirer de l'argent d'une assurance :	12
2.2	Use-case Application 2	12

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 et le serveur

Description : Permet au client de demander un devis pour un certain type 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'institution 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 concernant.
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 souscrit dans l'institution qui correspond au portefeuille sélectionné.

Préconditions : Le client doit avoir choisi un de ses portefeuilles 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 souscrit 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. Résilier une assurance
2. Souscrire à une assurance

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

2.1.4 Gérer les paramètres :

Acteurs : Le client et le serveur.

Description : Le client accède aux paramètres 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 clique sur le bouton des paramètres pour cette assurance.
5. L'application affiche l'onglet des paramètres pour assurance.
6. Le client effectue des modifications.
7. L'application envoie les modifications au serveur.
8. Le serveur effectue les modifications dans la base de donnée.
9. Le serveur envoie les nouvelles informations à l'application.
10. L'application affiche les nouvelles informations.
11. Le client quitte les paramètres.

Postconditions : La GUI affiche l'assurance avec les nouveaux paramètres qui lui ont été appliqués.

Parcours alternatif : Le client n'effectue pas de modifications et quitte la fenêtre.

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 client et le serveur.

Description : Le client décide depuis la liste de ses assurances de souscrire à une nouvelle assurance dans l'institution à laquelle son 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 redirigé 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 sa 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 résilier une assurance à laquelle il avait souscrit.

Préconditions : Le client doit être 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 résilier.

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 assurances 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 Effectuer une transaction :

Acteurs : Le client et le serveur.

Description :

Préconditions :

Fréquence :

Parcours de base :

Postconditions :

2.1.9 Payer la primer :

Acteurs :

Description :

Préconditions :

Fréquence :

Parcours de base :

Postconditions :

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

Acteurs :

Description :

Préconditions :

Fréquence :

Parcours de base :

Postconditions :

2.2 Use-case Application 2