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

1.1.1 Ré-initialiser le mot de passe

Acteurs :
Le client ou l'utilisateur et le serveur.

Description :
Permet à une personne qui a oublié son mot de passe ou qui souhaite le changer de le réinitialiser.

Précondition :
La personne (client ou institution) est dans l'écran de connection ou dans ses paramètres du compte.

Fréquence :
Assez rare

Parcours de base :

1. La personne clique sur l'option de ré-initialisation de mot de passe
2. La personne entre le mail du compte dont elle veut changer le mot de passe
3. L'application envoie le mail au serveur pour vérifier s'il existe un compte associé à l'adresse mail
4. Le compte est trouvé et un mail est envoyé avec un lien pour entrer un nouveau mot de passe
5. La personne entre 2 fois son nouveau mot de passe
6. Le serveur change le mot de passe du compte dans la base de données

Parcours alternatif :

1. Aucun compte avec ce mail n'est trouvé, message d'erreur et retour à l'étape 2.
2. La personne n'entre pas 2 fois le même mot de passe, retour à l'étape 5.

Postconditions :
Le mot de passe pour le compte associé au mail donné est changé par le nouveaux mot de passe entré par la personne.

1.1.2 Se déconnecter

Acteurs :
Le client et le serveur.

Description :
Permet au client de se déconnecter de l'application.

Précondition :
Le client est connecté à un compte.

Fréquence :
De temps en temps.

Parcours de base :

1. Le client demande à l'application de le déconnecter
2. L'application détruit l'objet utilisateur représentant le client authentifié
3. L'application ramène le client sur l'écran de connexion

Postconditions :
Le client est déconnecté et de retour à l'écran de connexion. Il n'a plus accès à ses données ni au reste de l'application.

1.1.3 Changer de langue

Acteurs :
Le client et le serveur.

Description :
Change la langue d'a chage et l'application.

Précondition :
L'application est lancée.

Fréquence :
Peu fréquent.

Parcours de base :

1. Le client sélectionne l'option de changement de langue
2. L'application a che les langues disponibles
3. Le client sélectionne la langue désirée
4. L'application a che un avertissement et demande une confirmation

5. Le client confirme le changement de la langue
6. L'application envoie une requête au serveur pour chnager la langue d'affichnage du client

Parcours alternatif :

1. Le client annule le changement de la langue, l'application renvoie sur le GUI les paramètres
2. Une erreur de connexion a eu lieu et la requête a échoué, pas de changement de la langue, l'utilisateur est averti.

Postconditions :

La langue d'a chage de l'application est changée.

1.2 Use-cases Application 1

1.2.1 Créer un compte

Acteurs :
Créer un compte.

Description :
Permet au client de créer un compte avec un mail et un mot de passe pour pouvoir utiliser l'application.

Précondition :
L'application est lancée et aucun compte n'est connecté.

Fréquence :
Chaque première fois qu'un nouveaux client lance l'application.

Parcours de base :

1. Le client clique sur l'option "créer un compte"
2. Le client entre le mail et le mot de passe qui ser associé au compte
3. L'application emvoie les identifiants au serveur pour vérification
4. Le serveur confirme que le mail n'est pas déjà pris
5. Le srveur crée le compte dans la base de données
6. Le serveur connecte automatiquement le nouveau client
7. Le client accède au reste de l'application avec le compte fraichement créé

Parcours alternatif :

1. Le serveur retourne que le mail est déjà associé à un compte existant. L'application le signale au client et le renvoie à l'étape 2

Postconditions :
Le client est connecté à l'application.

1.2.2 Se connecter

Acteurs :
Le client et le serveur.

Description :
Permet aux clients de s'identifier et d'accéder à l'application ainsi que toutes ses fonctionnalités.

Précondition :
L'application est lancée et le client n'est pas déjà connecté à l'pplication.

Fréquence :
A chaque lancement de l'application.

Parcours de base :

1. Le client entre ses données d'utilisateur (numéro registre national + mot de passe)
2. L'applicaton envoie les données au serveur pour la vérification
3. Le serveur confirme que les données sont correctes
4. Le client peut accéder aux autres parties de l'application

Parcours alternatif :

1. Les données sont incorrectes, on retourne à l'étape 1

Postconditions :
Le GUI a che désormais les di érents portefeuilles de l'utilisateur stockés dans la base de donnée.

1.2.3 Créer un portefeuilles

Acteurs :
Le client et le serveur.

Description :
Création d'un nouveau portefeuille vide avec le créateur comme gestionnaire et le seul membre du nouveau portefeuille.

Précondition :
Pas de préconditions.

Fréquence :
Utilisation peur fréquente.

Parcours de base :

1. Le client sélectionne l'option de créer un nouveau portefeuille
2. Le client choisit une institution financière (où il possède déjà un produit financier)
3. L'application envoie la demande au serveur
4. Le serveur confirme la demande

5. L'utilisateur est renvoyé sur le GUI "Wallet list"
6. Le serveur renvoie la demande sur l'application 2 de l'institution financière
7. L'institution financière confirme la demande de création de portefeuille
8. L'application 2 renvoie la confirmation au serveur

Parcours alternatif :

1. Une erreur de connexion à lieu, la requête échoue, l'utilisateur est averti, la créaton de portefeuille est annulée, l'utilisateur est renvoyé sur le GUI "Wallet list"
2. Le serveur refuse la demande, la création de portefeuille est annulée, le client est averti, l'utilisateur est renvoyé sur la GUI "Wallets list"
3. Une erreur de connexion à lieu , la requête échoue, l'utilisateur est averti, la création de portefeuille est annulée
4. L'institution financière refuse, l'utilisateur est notifié, la création de portfeuille est annulée
5. Une erreur de connexion à lieu, la requête échoue, l'utilisateur est averti, la création de portefeuille est annulée

Postconditions :
Un nouveau portefeuille a été créé / un nouveau portefeuille n'a pas été créé.

1.2.4 Supprimer le portefeuille

Acteurs :
Le client et le serveur.

Description :
Supprime le portefeuille sélectionné.

Précondition :
Le consommateur est un gestionnaire du portefeuille.

Fréquence :
Utilisation peur fréquente.

Parcours de base :

1. Le gestionnaire du portefeuille sélectionne l'option de suppression du portefeuille en question
2. L'application a che un avertissement et demande une confirmation
3. Le gestionnaire confirme la suppression

4. L'application envoie une requête au serveur de supprimer le portefeuille en question de la base de données et l'enlève de la liste de portefeuilles de tous les membres du portefeuille supprimé
5. L'application renvoi l'ancien gestionnaire vers sa liste de portefeuilles

Parcours alternatif :

1. L'utilisateur annule la suppression, l'application le renvoie vers la fenêtre du portefeuille en question
2. Une erreur de connexion a eu lieu et la requête a échoué, pas de suppression du portefeuille en question

1.2.5 Consulter portefeuille

Acteurs :
Le client et le serveur.

Description :
Le client accède au portefeuille et y retrouve tout ses produits financiers.

Précondition :
Le client est connecté à son compte et possède au moins un portefeuille.

Fréquence :
Très souvent.

Parcours de base :

1. Le client sélectionne le portefeuille à consulter
2. L'application fait une requête au serveur pour avoir les produits financiers dans le portefeuille
3. L'application a che les produits financiers qui sont dans le portefeuille

Parcours alternatif :

1. Une erreur de connexion a eu lieu et la requête a échoué, l'utilisateur est averti et retourne sur le GUI "Home page"

Postconditions :
L'application a che désormais le portefeuille que l'utilisateur veux consulter.

1.2.6 Supprimer un produit financier

Acteurs :
Client et serveur.

Description :
Supprime le dit produit financier.

Précondition :
Le client est connecté, un portefeuille contenant au moins un produit financier existe.

Fréquence :
Occasionnel.

Parcours de base :

1. L'utilisateur appuie sur le bouton de suppression correspondant au produit financier qu'il souhaite supprimer
2. L'application a che un message de confirmation
3. L'utilisateur accepte
4. L'application envoie la requête de suppression au serveur

Parcours alternatif :

1. L'utilisateur refuse, la fenêtre se ferme et aucune opération n'est e ectuée
2. Une erreur de connexion a eu lieu et la requête a échoué, l'utilisateur est averti et retourne la liste des produits financiers

Postconditions :
Le portefeuille ne contient plus le produit financier supprimé.

1.2.7 Souscrire un produit financier

Acteurs :
Client et serveur.

Description :
Ajoute un produit financier.

Précondition :
Le client est connecté et possède au moins un portefeuille.

Fréquence :
Occasionnel.

Parcours de base :

1. L'utilisateur appuie sur le bouton d'ajout

2. Le client choisit le type de produit financier
3. L'application a che un message de confirmation
4. L'utilisateur accepte
5. L'application envoie la requête d'ajout au serveur

Parcours alternatif :

1. L'utilisateur refuse, la fenêtre se ferme et aucune opération n'est e ectuée
2. Une erreur de connexion a lieu et la requête a échoué, l'utilisateur est averti

Postconditions :
Le portefeuille contient le produit financier ajouté.

1.2.8 A cher l'historique de transaction

Acteurs :
Client et serveur.

Description :
A che la liste de toutes les transactions e ectuées.

Fréquence :
Souvent.

Parcours de base :

1. L'utilisateur sélectionne l'option d'a chage de l'historique
2. L'application envoie une requête au serveur pour accéder a l'historique

Parcours alternatif :

1. Une erreur de connexion a lieu et la requête échoue, l'utilisateur est averti

Postconditions :
L'application a che l'historique de transaction.

1.2.9 E ectuer une transaction

Acteurs :
Client et serveur.

Description :
Le client peut e ectuer une transaction.

Précondition :
L'utilisateur est connecté.

Fréquence :
Souvent.

Parcours de base :

1. L'utilisateur sélectionne l'option de transaction
2. L'utilisateur choisit le compte du quel e ectuer la transaction
3. L'utilisateur entre le compte du receveur
4. L'utilisateur entre le montant de la transactions
5. L'utilisateur entre une communication (structurée ou non)
6. L'application a che un message de résumé de la transaction
7. L'utilisateur confirme en entrant son code
8. L'application envoie une requête au serveur pour la transactions
9. Le serveur retire l'argent sur le compte du client
10. Le serveur ajoute l'argent sur le compte du receveur

Parcours alternatif :

1. L'utilisateur se trompe de code, un message d'avertissement est a ché et il est invité a réessayer
2. Une erreur de connexion a lieu et la requête échoue, l'utilisateur est avertis et la transaction est annulée
3. Une erreur de connexion a lieu et la requête échoue, l'utilisateur est avertis la transaction est annulée

Postconditions :
La transaction est e ectuée et le montant est débité.

1.2.10 Ajouter un co-titulaire

Acteurs :
Client 1, client 2 et le serveur.

Description :
Le client 1 ajoute le client 2 comme co-titulaire sur un produit financiers.

Précondition :
Le client 1 est connecté , le client 2 possède un compte dans la même institution financière que le client 1.

Fréquence :
Occasionnel.

Parcours de base :

1. Le client 1 sélectionne le produit financier pour le quelle il souhaite ajouter un co-titulaire
2. Le client 1 entre les identifiants du client 2 (nom prénom et numéro de registre nationale)
3. L'application a che un message de confirmation
4. Le client 1 confirme
5. Le client 2 est avertie de la procédure
6. L'application a che un message de confirmation au client 2
7. Le client 2 confirme
8. L'application envoie la requête au serveur

Parcours alternatif :

1. Le client 1 refuse et il est renvoyé le GUI d'ajout de co-titulaire
2. Le client 2 refuse, la procédure est annulée
3. Une erreur de connexion a lieu et la requête échoue, l'utilisateur est averti et l'ajour du co-titulaire est annulée

Postconditions :
Le client 2 est ajouté en tant que co-titulaire du produit financier et peux maintenant le consulter et le modifer.

1.3 Use-cases Applications 2

1.3.1 Ajouter un client

Acteurs :
Employé d'une institution et serveur.

Description :
Ajoute un nouvau client.

Précondition :
L'employé de l ;institution est connecté.

Fréquence :
Souvent.

Parcours de base :

1. L'employé de l'institution appuie sur le bouton d'ajout de client
2. L'employé de l'institution entre les donnée du client (nom, prénom, numéro de registre national)
3. L'employé de l'institution entre les donnée du produit financier auquel le client souhaite souscrire
4. L'application envoie un message de confirmation
5. L'employé de l'institution accepte
6. L'application envoie la requête d'ajout au serveur

Parcours alternatif :

1. L'employé de l'institution refuse, le message se ferme et aucune opération n'est e ectuée et l'employé de l'institution est renvoyé sur la fenêtre d'ajout de client
2. Une erreur de connexion a lieu et la requête a échoué, l'employé de l'institution est averti et est renvoyé sur la fenêtre d'ajout de client

Postconditions :
Le client est ajouté comme clinet de l'institution financière.

1.3.2 Consulter la liste des clients et leurs produits financiers

Acteurs :
L'employé de l'instiution et le serveur.

Description :
Permet a l'employé de l'institution de pouvoir consulter la liste des clients de

l'institution financière chez qui l'employé travail ainsi que leurs produits financier.

Précondition :
L'employé de l'institution est connecté.

Parcours de base :

1. L'employé de l'institution appuie sur le bouton pour consulter la liste de clients ainsi que leurs produits financier
2. L'application envoie une requête au serveur
3. Le serveur reçoit la requête
4. Le serveur envoie les données nécessaire

Parcours alternatif :

1. Il y a un problème de connexion, l'employé de l'institution est averti et est renvoyé sur le menu de base

Postconditions :
L'application a che la liste de client et leurs produits financiers.

1.3.3 Activer les virements

Acteurs :
Employé d'une institution et serveur.

Description :
.

Précondition :
L'employé d'une institution est connecté et le client de l'institution a e ctué la demande d'activation de virement bancaire.

Fréquence :
.

Parcours de base :

1. L'employé d'une institution appuie sur le bouton pour consulter les demandes d'activation de virement
2. L'employé d'une institution accepte la demande du client
3. L'application envoie un message de confirmation
4. L'empployé d'une institution confirme

5. Les informations sont envoyées au serveur
6. Le serveur reçoit les informations

Parcours alternatif :

1. L'employé d'une institution refuse et il est renvoyé sur la fenêtre contenant les demandes d'activation de virement.
2. Il y a une problème de connexion, l'employé de l'institution est averti et est renvoyé sur la fenêtre contenant les demandes d'activation de virement

Postconditions :
Les virements sont autorisés pour le client.

1.3.4 Supprimer un client

Acteurs :
Employé d'une institution et serveur.

Description :
Supprime définitivement un client de l'institution.

Précondition :
L'employé d'une institution est connecté, le client est client chez l'institution de l'employé.

Fréquence :
Occassionel.

Parcours de base :

1. L'employé d'une institution appuie sur le bouton de suppression
2. L'employé d'une institution entre les données du client à supprimer
3. L'application a che un message de confirmation
4. L'employé d'une institution accepte
5. L'application envoie la requête de suppression au serveur

Parcours alternatif :

1. L'employé d'une institution e ectue un clique droit sur un client puis l'application a che des options supplémentaires, l'employé d'une institution sectionne la suppression du client, l'application a che un message de confirmation, l'employé d'une institution confirme, la requête est envoyé au serveur
2. L'employé d'une institution refuse, la fenêtre se ferme et aucune opération n'est e ectué

3. Une erreur de connexion a lieu et la requête a échoué, l'employé d'une institution et averti et retourne la liste des clients

Postconditions :
Le client est supprimé et ne figure plus dans la liste des clients.

1.3.5 Supprimer un produit financier

Acteurs :
Employé d'une institution et serveur.

Description :
Supprime définitivement un prodiut financier d'un client.

Précondition :
L'employé d'une institution est connecté, le client est client chez l'institution de l'employé.

Fréquence :
Occassionel.

Parcours de base :

1. L'employé d'une institution appuie sur le bouton de suppression
2. L'employé d'une institution entre les données du client et du produit à supprimer
3. L'application a che un message de confirmation
4. L'employé d'une institution accepte
5. L'application envoie la requête de suppression au serveur

Parcours alternatif :

1. L'employé d'une institution e ectue un clique droit sur un client puis l'application a che des options supplémentaires, l'employé d'une institution sectionne la suppression du produit, l' employé sélectionne le produit à supprimer l'application a che un message de confirmation, l'employé d'une institution confirme, la requête est envoyé au serveur
2. L'employé d'une institution refuse, la fenêtre se ferme et aucune opération n'est e ectué
3. Une erreur de connexion a lieu et la requête a échoué, l'employé d'une institution est averti et retourne la liste des clients

Postconditions :
le produit financier est supprimé et ne figure plus dans la liste du client qui le possédait.

1.3.6 Ajouter un produit financier

Acteurs :
Employé d'une institution et serveur.

Description :
Ajoute un produit financier à un client.

Précondition :
L'employé d'une institution est connecté le client possède un compte chez l'institution.

Fréquence :
Souvent.

Parcours de base :

1. L'employé d'une institution appuie sur le bouton d'ajout
2. L'employé d'une institution entre les données du client et du produit à ajouter
3. L'application a che un message de confirmation
4. L'employé d'une institution accepte
5. L'application envoie la requête de suppression au serveur

Parcours alternatif :

1. L'employé d'une institution e ectue un clique droit sur un client puis l'application a che des options supplémentaires, l'employé d'une institution sectionne l'ajout d'un produit, l' employé entre les données du produit à ajouter l'application a che un message de confirmation, l'employé d'une institution confirme, la requête est envoyé au serveur
2. L'employé d'une institution refuse, la fenêtre se ferme et aucune opération n'est e ectué
3. Une erreur de connexion a lieu et la requête a échoué, l'employé d'une institution est averti et retourne la liste des clients

Postconditions :
Le produit financier est ajouté et figure dans la liste du client qui le possède.

1.3.7 Trier les produits financiers

Acteurs :
Employé d'une institution.

Description :
L’employé d’une institution trie la liste des produits financiers.

Précondition :
L’employé d’une institution est connecté.

Fréquence :
Souvent.

Parcours de base :

1. L’ employé d’une institution sélectionne en fonction de quoi il souhaite trier les produis financier
2. L’application trie les produits financier

Postconditions :
La liste est triée.

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 a che le devis que le client a demandé.

2.1.2 Obtenir des informations sur les di érents 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 a che 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 a che 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'a cher également les assurances qu'il a résilié. Et le serveur renvoie également les assurances qui sont désactivées afin de les a cher.

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 a che 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'a chage 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 a che l'historique de l'assurance du client
5. Le client consulte son historique

Postconditions : La GUI a che 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 e ectuer 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 su samment de liquidités sur le compte sélectionné.
7. Le serveur e ectue 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 a che la nouvelle assurance dans la liste.

Postconditions : La GUI a che la nouvelle assurance dans la liste des paiements ainsi que la prochaine date de paiement. La GUI met également à jour l'a chage 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 insu sant.

4. L’application a che 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 e ectue 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 a che bien que l'assurance est désactivée si le client a che les assurance désactivées et n'a che 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 a che 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 e ectuer 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 e ectuer 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 su sant.
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'a chage.
8. Le client reçoit la confirmation du paiement.

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

Parcours alternatif :

1. Le serveur vérifie le solde et celui-ci est insu sant.
2. Le serveur envoie une erreur à l'application stipulant que le solde du compte sélectionné est insu sant.
3. Si le client est connecté l'application lui a che 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'a chage en fonction des informations renvoyées.

Postconditions : La GUI a che les modifications qui ont été e ectuées et notifie l'utilisateur que l'opération a bien été e ectuée.

Parcours alternatif :

1. Après vérification le serveur détecte qu'un solde n'est pas su sant.
2. Le serveur envoie l'erreur à l'application et annule l'opération.
3. L'application a che 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 e ectué 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 a che 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 institution.

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 :

4.b L'utilisateur entre le mauvais code pin. Il est invité à entrer son code de nouveau.

6.b L'utilisateur entre le mauvais code d'autentification. Il est invité à entrer son code d'autentification

Postconditions :Le client est connecté.

3.1.2 Demander une carte

Acteurs : Le client, le serveur et un employé de l'institution financière

Description : Le client fait une damande de carte et un employé de l'institution va valider cette demande.

Préconditions : L'utilisateur est client de l'institution dans laquelle il fait sa demande et est connecté à l'application.

Fréquence :très rarement

Parcours de base :

1. Le client choisit le compte de paiement pour lequel il veut recevoir une carte.
2. Le client choisit le type de carte qu'il veut recevoir.
3. La demande est envoyé au serveur.
4. Un employé de l'institution valide la demande.

Parcours alternatif :

1.b Le client crée le compte pour lequel il veut recevoir une carte.
4.b L'employée refuse la demande du client.

Postconditions : Le client recoit sa carte de paiement.

3.1.3 Bloquer une carte

Acteurs : Le client et le serveur

Description : Le client choisit de bloquer une de ses cartes.

Préconditions : L'utilisateur est connecté à l'application et posède une carte.

Fréquence : très rarement

Parcours de base :

1. L'utilisateur sélèctionne une de ses cartes.
2. L'utilisateur choisit de bloquer la carte sélectionnée
3. L'application demande à l'utilisateur de confirmer son choix
4. L'utilisateur confirme
5. L'application envoie la demande au serveur

Parcours alternatif :

3.b L'utilisateur annule sa décision et revient sur le menu de la carte

Postconditions : La carte de l'utilisateur est bloquée, il ne peut plus l'utiliser

3.1.4 Modifier les paramètres d'une carte

Acteurs : Client et serveur

Description :Le client fait des modifications sur sa carte qui seront envoyées au serveur.

Préconditions : Le client possède une carte et est connecté à l'application

Fréquence :régulièrement

Parcours de base :

1. Le client entre dans les menu des paramètres de sa carte
2. Le client choisit les paramètres qu'il veut modifier
3. À la confirmation, les nouvelles valeurs des paramètres sont envoyées au serveur

Postconditions : Les changements ont été appliqués

3.1.5 confirmer le renouvellement

Acteurs : Client et serveur

Description :Le client active le renouvellement automatique de sa carte lorsque celle-ci dépasse sa date limite d'utilisation

Préconditions : Le client possède une carte et est connecté à l'application

Fréquence :très rarement

Parcours de base :

1. L'utilisateur active l'option de renouvellement
2. L'utilisateur confirme
3. L'application envoie au serveur la modification de l'option de renouvellment

Parcours alternatif :

2.b L'utilisateur annule l'activation du renouvellement automatique

Postconditions : Le renouvellement automatique de la carte est activé

3.1.6 Consulter/ exporter l'historique d'utilisation d'une carte

Acteurs : Client et serveur

Description : Le client a che ou exporte l'historique d'utilisation de sa carte

Préconditions : L'utilisateur possède une carte et est connecté à l'application

Fréquence : souvent

Parcours de base :

1. L'utilisateur entre dans le menu historique de sa carte
2. L'application récupère les données depuis le serveur
3. L'utilisateur choisit d'exporter ses données
4. L'application génère un fichier json contenant les données de l'historique de la carte

Parcours alternatif :
2.b L'utilisateur n'a pas de données pour la carte sélectionnée

4 Extension 6 : Virements et gestions de fraudes - Agbenda Pignozi

4.1 Use-case Application 1

4.1.1 Demande de paiement par code QR

Acteurs : Client d'une institution

Description : Le client génère un code QR qui, une fois scanné, permettra d'e ectuer une transaction

Précondition : Le clien est connecté à l'application

Fréquence : Occasionnelle

Parcours de base :

1. Le client clique sur le bouton QR code.
2. Le client choisit l'option « create payment request ».
3. Le client entre le numéro du compte bancaire sur le quel sera versé l'argent.
4. Le client entre le montant de la transaction.
5. Le client entre un message qui sera a ché quand la transaction sera e ectuée.
6. Le client clique sur bouton « generate ».
7. L'application génère un code QR et le stock localement.

Parcours alternatif :

Postcondition : Le code QR est généré.

4.1.2 Payement par code QR

Acteurs : Client d'une institution et le serveur

Description : Le client e ectue une transaction en scannant un code QR.

Précondition : Le clien est connecté à l'application.

Fréquence : Occasionnelle

Parcours de base :

1. Le client clique sur le bouton QR code.
2. Le client choisit l'option « load payment request ».
3. Le client entre le numéro du compte bancaire sur le quel l'argent sera débité.
4. Le client clique sur le bouton «load».
5. Le client choisit le code QR à scanner.
6. Le client confirme le scan.
7. Le l'application envoie la requête au serveur.
8. Le serveur e ectue la requête.

Parcours alternatif :

7.b Une erreur de connexion a lieu et la requête a échoué, le client est averti, il est redirigé vers le menu de payement par code QR

Postcondition : La transaction est e ectuée.

4.1.3 E ecteur un payement avec/sans contact

Acteurs : Client d'une institution et le serveur

Description : Le client e ectue un virement rapide avec/sans contact

Précondition : Le client est connecté à l'application

Fréquence : Souvent

Parcours de base :

1. Le client sélectionne l'option de payement rapide.
2. Le client entre le compte qui sera débité.
3. Le client sélectionne la transaction qu'il souhaite e ectuer.
4. Le client choisit le payement avec contact.
5. Le client entre le code PIN du compte bancaire.
6. Le client confirme la transaction.
7. Le l'application envoie la requête au serveur.
8. Le serveur e ectue la requête.

Parcours alternatif :

4.b Le client choisit le payement sans contacte, la transaction est e ectué

5.b Le client se trompe de code PIN, il ne lui reste plus que 2 chances pour entrer le bon code PIN

8.b Une erreur de connexion a lieu et la requête a échoué, le client est averti, il est redirigé vers le menu de payement rapide

Postcondition : Le virement est e ectué.

4.1.4 Gérer les limites des payement avec/sans contact

Acteurs : Employé d'une institution financière et le serveur

Description : L'employé de l'institution financière peux changer le plafond limite (quotidien, hebdomadaire, mensuel) pour les payements avec ou sans contact

Précondition : L'employé est connecté

Fréquence : Occasionnelle

Parcours : de base

1. L'employé clique sur le bouton pour définir les plafonds.
2. L'employé entre les di érents plafonds.
3. L'employé confirme les plafonds.
4. L'application envoie la requête au serveur.
5. Le serveur e ectue la requête.

Parcours alternatif :

2.b L'employé coche l'option pour ne pas mettre de plafonds, aucun plafond n'est appliqué.

5.b Une erreur de connexion a lieu et la requête a échoué, l'employé est averti, il est redirigé vers le menu de définition de plafonds.