Stratégie de test

Scénarios prévus

- Créer un compte utilisateur
- S’authentifier dans l’application
- Se déconnecter
- Récupérer les données bancaires
- Consulter les données bancaires
- Télécharger les relevés de compte
- Consulter une synthèse des dépenses
- Prendre rendez-vous avec un conseiller

Méthodes de test adaptées

- **Créer un compte utilisateur :**
  - **Tests automatisés** pour tout ce qui concerne le formulaire d’inscription parce qu’il y a beaucoup de déclinaisons sur chacun des champs du formulaire. L’automatisation n’est pas complexe à développer sur ces tests et permettra de gagner du temps sur leur exécution, qui sera d’ailleurs plus fréquente sur cette fonctionnalité critique de l’application.
  - **Tests manuels** pour tout ce qui concerne l’écran « Vos justificatifs » car ces derniers requièrent non seulement l’utilisation d’un système externe, mais également la capacité à reconnaître des patterns dans des documents PDFs, ce qui rend l’automatisation des tests bien plus complexe et moins rentable à mettre en place.

- **S’authentifier dans l’application :**
  - **Tests automatisés** pour tout ce qui concerne le formulaire de connexion, pour les mêmes raisons que le formulaire d’inscription. Ainsi que pour tout ce qui concerne les appels API. L’automatisation des tests d’API est justifiée premièrement par la simplicité de sa mise en place ainsi que la rapidité de l’éxécution des tests, et deuxièmement par le besoin de s’assurer régulièrement et facilement de la bonne intégration avec l’API et de l’intégrité des données qu’elle nous renvoie.
  - **Tests manuels** pour tout ce qui concerne le mot de passe oublié car ces tests requièrent l’utilisation d’un système externe, ce qui rend leur automatisation plus coûteuse à mettre en place.

- **Tests en boîte noire** pour tout ce qui concerne l'authentification biométrique car il est impossible d'automatiser ces tests, et que le système de reconnaissance de l'empreinte digitale dépend de l'OS du périphérique utilisé par l'utilisateur.

- **Se déconnecter :**
  - **Tests automatisés** car ils présentent un grand nombre d'avantages sur cette fonctionnalité critique de l'application. Premièrement, ils sont très faciles à mettre en place ici. Deuxièmement, ils permettront de s'assurer facilement et rapidement que l'utilisateur peut toujours se déconnecter, et donc rendre l'accès à ses informations sensibles impossible à des utilisateurs malveillants, garantissant ainsi une plus grande sécurité de ses données. Pour finir, certains de ces tests requièrent une attente de cinq minutes, ce qui représente une quantité de temps non négligeable si une personne devait effectuer le test manuellement.

- **Récupérer les données bancaires :**
  - **Tests automatisés** car il ne s'agit dans ce scénario que de tests d'API. L'automatisation des tests d'API se justifie ici de la même façon qu'elle se justifie pour l'authentification. C'est d'autant plus important lorsqu'il s'agit d'une API externe telle que celle de la Banque de France.

- **Consulter les données bancaires :**
  - **Tests manuels** sur l'ensemble du scénario car les tests ne font que valider l'interface utilisateur. La simplicité d'exécution de ces tests rend un investissement dans leur automatisation non rentable.

- **Télécharger les relevés de compte :**
  - **Tests manuels** opérés sur le point de terminaison de l'API de la Banque de France qui permet de récupérer les pdfs des relevés de compte. Certains de ces tests requérant la capacité à reconnaître des patterns dans un document pdf, leur exécution par un membre de l'équipe est plus rapide et moins coûteuse à mettre en place.

- **Consulter une synthèse des dépenses :**
  - **Tests manuels** sur l'ensemble du scénario pour la même raison que sur le scénario de consultation des données bancaires.
  - **Tests exploratoires** à effectuer en plus des tests manuels sur

la fonctionnalité de paramétrage de la périodicité de la synthèse des dépenses, afin de couvrir plus de cas de tests limites (edge cases) sur cette dernière.

- **Prendre rendez-vous avec un conseiller**
  - **Tests manuels** sur l'ensemble du scénario pour la même raison que sur les scénarios de consultation des données bancaires et de consultation de la synthèse des dépenses.

Ressources nécessaires

- **L'environnement de test :**
  - Un sous-domaine de test **test.tomsen.fr** pemettant de mieux simuler les conditions d'un environnement réel sans risquer d'impacter les données des utilisateurs finaux. Il nous sera utile pour simuler de façon plus réaliste les appels aux différents points de terminaison de l'API de la Banque de France.
  - Une base de données de test

- **Les données de test :**
  - Au moins 1 compte utilisateur (Nom, Prénom, Date de naissance, Mot de passe, Justificatif de domicile, Relevé d'imposition, Carte d'identité) + données et fichiers invalides
  - Au moins 2 comptes bancaires, 1 courant et 1 épargne (Identifiant client, Numéro de compte)
  - Au moins 5 transactions (Raison sociale des destinataires, Montant, Date)
  - Au moins deux conseillers épargne (Nom, Date et Heure de disponibilité)
  - Au moins deux conseillers consommation (Nom, Date et Heure de disponibilité)

- **Les ressources humaines :**
  - 1 testeur pour exécuter les test manuels, d'exploration, en boîte noire, et automatisés, analyser les résultats des tests, rapporter et suivre les anomalies, rédiger et présenter le bilan de la campagne de tests.
  - 1 développeur pour rédiger le programme d'automatisation des tests définis

- **Les compétences :**
  - Maîtrise de Cypress pour l'automatisation des tests

    - Bonne compréhension du fonctionnement de l'interface graphique de Cypress pour l'exécution et l'analyse des résultats des tests automatisés
    - Maîtrise de Postman pour l'exécution manuelle de certains des tests d'API
    - Bonne compréhension du fonctionnement des API REST
    - Connaissance basique de SQL pour la gestion de la base de données de test
    - Bonne connaissance de JIRA pour le rapport des anomalies et le suivi de leur résolution
    - Connaissance basique de la suite Microsoft Office pour la rédaction et présentation du bilan de campagne de test

- **Les outils de test :**
    - Le framework **Cypress** pour l'automatisation des tests, et l'exécution des tests automatisés
    - L'éditeur de code **Visual Studio Code** pour la rédaction du programme d'automatisation des tests
    - L'application **Postman** pour exécuter les tests manuels d'API et analyser leurs résultats
    - Le système de gestion de base de données relationnelles **PostgreSQL** pour la gestion de la base de données de test
    - **Un périphérique doté d'une fonctionnalité d'authentification biométrique** pour tester cette dernière dans l'application
    - Le logiciel **Adobe Acrobat** pour visionner les pdfs de relevé de compte et procéder à la validation de leur conformité
    - L'application **JIRA** pour le rapport des anomalies et le suivi de leur résolution
    - La suite **Microsoft Office** pour la rédaction et la présentation du bilan de campagne de test

Étapes clés de la stratégie

- **Sprint 1 : 3h10**
    - Analyse des exigences si une nouvelle exigence est ajoutée en cours de route
    - Conception des scripts de tests automatisés :
        - Créer un compte utilisateur **2h**
    - Exécution du scénario de test :
        - Créer un compte utilisateur **10 minutes**
    - Analyse des résultats et rapport des anomalies s'il y en a **1h**
    - Suivi des anomalies si besoin

- Validation

- **Sprint 2 : 3h17**
  - Analyse des exigences si une nouvelle exigence est ajoutée en cours de route
  - Conception des scripts de tests automatisés :
    - S'authentifier dans l'application **1h50**
    - Se déconnecter **10 minutes**
  - Exécution des scénarios de test :
    - S'authentifier dans l'application **10 minutes**
    - Se déconnecter **5 minutes**
  - Analyse des résultats et rapport des anomalies s'il y en a **1h**
  - Suivi des anomalies si besoin
  - Validation

- **Sprint 3 : 2h27**
  - Analyse des exigences si une nouvelle exigence est ajoutée en cours de route
  - Conception des scripts de tests automatisés :
    - Récupérer les données bancaires **1h**
  - Exécution du scénario de test :
    - Récupérer les données bancaires **2 minutes**
  - Exécution des scénarios de non-régression :
    - Créer un compte utilisateur **10 minutes**
    - S'authentifier dans l'application **10 minutes**
    - Se déconnecter **5 minutes**
  - Analyse des résultats et rapport des anomalies s'il y en a **1h**
  - Suivi des anomalies si besoin
  - Validation

- **Sprint 4 : 1h28**
  - Analyse des exigences si une nouvelle exigence est ajoutée en cours de route
  - Exécution des scénarios de test :
    - Consulter les données bancaires **1 minute**
    - Télécharger les relevés de compte **2 minutes**
  - Exécution des scénarios de non-régression :
    - Créer un compte utilisateur **10 minutes**
    - S'authentifier dans l'application **10 minutes**
    - Se déconnecter **5 minutes**
  - Analyse des résultats et rapport des anomalies s'il y en a **1h**
  - Suivi des anomalies si besoin
  - Validation

- **Sprint 5 : 1h37**
  - Analyse des exigences si une nouvelle exigence est ajoutée en cours de route
  - Exécution du scénario de test :
    - Consulter une synthèse des dépenses **12 minutes**
  - Exécution des scénarios de non-régression :
    - Créer un compte utilisateur **10 minutes**
    - S'authentifier dans l'application **10 minutes**
    - Se déconnecter **5 minutes**
  - Analyse des résultats et rapport des anomalies s'il y en a **1h**
  - Suivi des anomalies si besoin
  - Validation

- **Sprint 6 : 2h28**
  - Analyse des exigences si une nouvelle exigence est ajoutée en cours de route
  - Exécution du scénario de test :
    - Prendre rendez-vous avec un conseiller **3 minutes**
  - Exécution des scénarios de non-régression :
    - Créer un compte utilisateur **10 minutes**
    - S'authentifier dans l'application **10 minutes**
    - Se déconnecter **5 minutes**
  - Analyse des résultats et rapport des anomalies s'il y en a **1h**
  - Suivi des anomalies si besoin
  - Validation
  - Bilan de campagne **1h**

L'analyse des exigences, si elle a lieu d'être, sera systématiquement effectuée en début de sprint, avant le développement des fonctionnalités.

La conception des scripts de test automatisés pourra être effectuée en même temps que le développement de la fonctionnalité, dans une approche de Test Driven Development.

L'exécution du scénario de test sera effectuée juste après le développement de la fonctionnalité à tester.

Les scénarios de non-régression seront exécutés après le développement des fonctionnalités, en fin de sprint.

Préconisations

- **Les éléments d'alerte :**
  - La carte d'identité que l'utilisateur va entrer dans l'application au moment de son inscription est une donnée très sensible qu'il convient de gérer correctement. **Elsa** devrait **entrer en contact avec la CNIL** afin de déterminer et assurer la bonne gestion de cette donnée dans l'application **avant le lancement du premier sprint**.
  - Le développement de l'application venant seulement d'être validé, la gestion des données n'a pas encore été prise en charge. Il conviendrait pour **Alice** de **créer l'architecture de la base de données avant le lancement du premier sprint**.
  - Le développement de l'application venant seulement d'être validé, il n'existe encore aucun environnement de test dédié. La **mise en place d'un sous-domaine de test par notre développeur back-end sous la supervision d'Alice** paraît essentiel **avant le lancement du premier sprint**.
  - La mise en place de tests pour la fonctionnalité d'**authentification biométrique** est très **complexe et coûteuse** à mettre en place. Elle nécessite d'être en possession d'un périphérique doté d'une fonctionnalité d'authentification biométrique et d'installer plusieurs systèmes d'exploitation sur ce même périphérique. Cela peut rapidement devenir cher et chronophage. Pour cette raison, la **couverture des tests** sur cette fonctionnalité de l'application devrait être **minimale**.

- **Les points clés à retenir :**
  - Aucun patrimoine de test n'existe encore pour l'application Tomsen. Un **cahier de recette** réunissant des cas de tests pour chaque scénario identifié dans la présente stratégie de test devrait être **rédigé par mes soins avant le lancement du premier sprint**.
  - Certaines données de test nécessitent un **partenariat avec la Banque de France**, notamment pour la réception de données de comptes bancaires et de transactions, ce afin d'obtenir une simulation la plus fidèle possible au comportement de l'application pour les utilisateurs finaux. **Quentin devrait contacter la Banque de France** pour mettre en place ce partenariat, définir les besoins en données de test et assurer leur adéquation avec le projet, **avant la conception des tests automatisés du troisième sprint**.
  - Certains scénarios de test comprennent des **scripts de test**

automatisés. Pour chaque sprint, les scripts des scénarios qui seront testés durant le sprint devraient être **rédigés par notre développeur front-end avant la phase d'exécution des tests**.

- Les fonctionnalités des trois premiers scénarios, à savoir la création d'un compte utilisateur, l'authentification dans l'application et la déconnexion, étant des **fonctionnalités critiques**, les **tests** de ces derniers devraient être **exécutés par mes soins à chaque sprint** afin d'assurer la non-régression de ces fonctionnalités. Ils pourront également servir de **tests de non-régression** lors de l'implémentation future d'autres fonctionnalités par d'autres équipes.