**Sujet de TD/TP 1 : Identification des Acteurs, Génération du Backlog et des Sprints, et Création de Diagrammes UML**

**Objectif :**

Dans ce TD/TP, vous allez explorer les concepts de **systèmes** et de **systèmes d'information** en vous appuyant sur la plateforme **Wave** . Vous travaillerez sur l'identification des acteurs, la génération d'un backlog, et l'organisation des fonctionnalités en sprints. Vous modéliserez également le système à l'aide de diagrammes UML, en mettant en évidence les cas d'utilisation et la structure du système.

**Étapes de la mission :**

**1. Identification des Acteurs :**

Les **acteurs** dans le cadre d'un système d'information représentent les entités qui interagissent directement ou indirectement avec le système.

**Exemples d'acteurs pour la plateforme Wave :**

* **Utilisateur (Client)** : Une personne utilisant la plateforme pour effectuer des transactions (envoyer ou recevoir de l'argent).
* **Agent Wave** : Une personne physique qui permet aux utilisateurs de déposer ou de retirer de l'argent.
* **Administrateur du Système** : Une personne responsable de la gestion des serveurs, de la sécurité et de la maintenance de la plateforme.
* **Partenaires Commerciaux** : Entreprises ou services utilisant Wave pour recevoir des paiements (ex. services publics).
* **API Bancaires** : Systèmes externes pour l'intégration avec les banques locales.

**2. Définir des personnes :**

Les **personas** sont des profils fictifs d'utilisateurs représentant les différents types d'acteurs. Ils permettent de mieux comprendre les attentes et besoins des utilisateurs.

**Exemples de personnes :**

* **Aïcha, Utilisatrice Rurale** : Aïcha est une commerçante vivante dans une zone rurale. Elle utilise Wave pour envoyer de l'argent à sa famille et payer ses fournisseurs. Elle se rend souvent chez l'agent Wave pour déposer et retirer de l'argent.
* **Moussa, Agent Wave** : Moussa gère un kiosque Wave dans une zone urbaine et aide les utilisateurs à déposer et retirer de l'argent. Son principal souci est d'assurer la rapidité et la sécurité des transactions.
* **Fatou, Partenaire Commercial** : Fatou est propriétaire d'une petite entreprise qui utilise Wave pour recevoir des paiements électroniques pour ses ventes.

**3. Générer un Backlog :**

Le **backlog** est une liste de fonctionnalités à développer pour la plateforme Wave. Chaque fonctionnalité doit être définie clairement et priorisée en fonction des besoins.

**Exemples de fonctionnalités à inclure dans le backlog :**

* **Créer un compte utilisateur** : Permettre aux nouveaux utilisateurs de s'inscrire sur la plateforme.
* **Effectuer un dépôt/Retrait** : Permettre aux utilisateurs de déposer ou de retirer de l'argent via un agent Wave.
* **Envoyer de l'argent à un autre utilisateur** : Fonction essentielle de transfert d'argent entre utilisateurs Wave.
* **Consulter l'historique des transactions** : Permettre aux utilisateurs de voir leurs transactions passées.
* **Notifier les utilisateurs par SMS lors des transactions** : Envoi automatique de notifications après chaque opération réussie.
* **Système de gestion des agents** : Interface dédiée à la gestion des agents pour assurer une couverture nationale efficace.

**4. Organisation du Backlog en Sprints :**

Une fois les fonctionnalités répertoriées, elles doivent être réparties en **sprints** selon les priorités et les délais. Les sprints représentent des cycles de développement courts (souvent de 1 à 2 semaines) durant lesquels certaines fonctionnalités sont développées.

**Exemples d'organisation :**

* **Sprint 1** (2 semaines) :
  + Créer un compte utilisateur.
  + Effectuer un dépôt/retrait.
* **Sprint 2** (2 semaines) :
  + Envoyer de l'argent à un autre utilisateur.
  + Consulter l'historique des transactions.
* **Sprint 3** (2 semaines) :
  + Notifier les utilisateurs par SMS lors des transactions.
  + Système de gestion des agents.

**5. Création des Diagrammes UML :**

Les **diagrammes UML** permettent de modéliser les interactions entre les acteurs et le système, ainsi que la structure interne du système.

**Diagrammes à produire :**

* **Diagramme de cas d'utilisation** : Ce diagramme met en évidence les **interactions** entre les acteurs et le système. Chaque fonctionnalité du backlog peut être représentée par un cas d'utilisation.

Exemples de cas d'utilisation :

* + "Créer un compte utilisateur" (acteur : utilisateur).
  + "Effectuer un dépôt/retrait" (acteur : utilisateur, agent Wave).
  + "Consulter l'historique des transactions" (acteur : utilisateur).
* **Diagramme de classes** : Le **diagramme de classes** représente la **structure du système** , les différentes entités (classes) qui composent le système et leurs relations.

Exemples de classes à inclure :

* + **Utilisateur** : Attributs (nom, numéro de téléphone, solde) ; Méthodes (créer un compte, consulter solde).
  + **Agent Wave** : Attributs (nom, ID agent) ; Méthodes (effectuer un dépôt, effectuer un retrait).
  + **Transaction** : Attributs (montant, date, type) ; Méthodes (enregistrer la transaction).

**Réflexion sur les Enjeux, Défis et Objectifs :**

* **Enjeux** : Assurer la sécurité et la fiabilité des transactions financières. Offrir une solution accessible aux utilisateurs non bancarisés.
* **Défis** : Gestion des photos de charge réseau, protection des données personnelles, intégration avec des systèmes bancaires externes.
* **Risques** : Pannes du système, cyberattaques, mauvaise expérience utilisateur.
* **Objectifs** : Fournir une plateforme robuste, sécurisée et facile à utiliser, qui permet d'améliorer l'inclusion financière.

**Méthodologie :**

Pendant le TD/TP, vous serez guidé à travers ces étapes. Vous devrez utiliser des outils comme **StarUML** , **Draw.io** , ou toute autre solution de modélisation UML, ainsi qu'un outil de gestion de projet pour organiser le backlog (par exemple **Trello** , **Jira** , ou **Azure DevOps** ).

**Déroulement :**

* Introduction et discussion (20 minutes).
* Travail en groupe sur l'identification des acteurs et la création des personas (40 minutes).
* Travail sur la génération du backlog et l'organisation des sprints (1 heure).
* Création des diagrammes UML (1h30).
* Présentation des résultats par chaque groupe (30 minutes).

**Livrables :**

1. Liste des **acteurs** et **personas** .
2. **Backlog** priorisé et organisé en sprints.
3. **Diagramme UML** : Cas d'utilisation et Diagramme de classes.

**Conclusion :**

Ce TP vous permettra de mieux comprendre la **modélisation des systèmes d'information** à travers l'utilisation des méthodes agiles et UML. Vous apprendrez à identifier les acteurs clés, à organiser un backlog, et à structurer un système complexe comme la plateforme Wave, tout en prenant en compte les défis et enjeux inhérents au développement de systèmes d'information.

Voici 5 modules clés du système Wave, chacun accompagné de quelques fonctionnalités principales :

**1. Module de Gestion des Utilisateurs**

* **Création et gestion de compte utilisateur** : Permet aux utilisateurs de créer des comptes, de définir leurs informations personnelles et de gérer leurs profils.
* **Vérification d'identité** : Utiliser des mécanismes d'authentification (codes PIN, SMS) pour garantir la sécurité des transactions et l'accès sécurisé au compte.
* **Gestion des niveaux de compte** : Offre différents niveaux de comptes en fonction des besoins (utilisateur standard, commerçant, agent).

**2. Module de Transfert d'Argent**

* **Transfert d'argent peer-to-peer (P2P)** : Permet aux utilisateurs d'envoyer et de recevoir de l'argent instantanément via leur compte Wave.
* **Notifications en temps réel** : Envoie des alertes par SMS ou notifications push à la fois pour l'expéditeur et le destinataire dès que le transfert est effectué.
* **Historique des transactions** : Les utilisateurs peuvent consulter un récapitulatif détaillé de tous les transferts d'argent effectués ou reçus.

**3. Module de Paiement Mobile**

* **Paiement de factures** : Permet de payer des factures (électricité, eau, services publics) directement depuis l'application.
* **Paiement dans les commerces partenaires** : Les utilisateurs peuvent régler leurs achats dans les magasins et auprès des commerçants affiliés à Wave.
* **Paiement récurrent** : Fonctionnalité permettant d'automatiser les paiements pour des services récurrents (abonnements, factures mensuelles).

**4. Module de Gestion des Agents**

* **Enregistrement et gestion des agents** : Permet aux agents Wave de s'inscrire et de gérer leurs profils, y compris les commissions perçues et le suivi des transactions.
* **Dépôt et retrait d'espèces** : Les agents peuvent faciliter les dépôts et retraits d'argent pour les utilisateurs Wave.
* **Gestion des commissions** : Suivi automatique des commissions gagnées par les agents sur chaque transaction.

**5. Module de Sécurité et Surveillance**

* **Gestion des authentifications** : Sécurise les connexions et transactions via des méthodes d'authentification fortes, comme les codes PIN ou biométriques (pour les smartphones compatibles).
* **Surveillance des fraudes** : Détection et prévention des comportements frauduleux en surveillant les transactions suspectes ou les anomalies.
* **Gestion des accès et permissions** : Définition des rôles et niveaux d'accès pour les différents utilisateurs, agents, et administrateurs.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Modules | Acteurs | Fonctionnalités | Règle de gestion | Risques | Opportunités | kpi | Integration avec un système exterieur |
| Gestion utilisateur | Administrateur systeme |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |