1. **Analyse, Conception et réalisation**
   1. **Spécification des besoins**
      1. **Besoins fonctionnels**

Le but d’un projet est de satisfaire un besoin. Il faut l’exprimer clairement avant d’imposer une solution.

Généralement formulé sous formes d’exigences fonctionnelles, les besoins fonctionnels sont l’expression de ce que le produit ou le service délivré par le projet devrait être ou faire.

Les besoins fonctionnels classés par utilisateur du système.

**Le système doit permettre à un investisseur :**

* De s’inscrire à travers un formulaire ;
* De se connecter avec ses identifiants ;
* De voir l’ensemble des projets auquel il participe ;
* Dans chacune de ses projets il pourra consulter la liste de toutes les transactions qu’il a effectué ;
* De faire un virement à travers les applications mobile money ou les banques pour investir

**Le système doit permettre à l’administrateur de la plateforme :**

* De se connecter ;
* D’ajouter des projets ;
* De valider les inscriptions de chaque utilisateur ;
* De faire les virements aux différents investisseur des projets ;
* De voir l’ensemble des projets de la plateforme ;
* De voir la liste de tous les investisseur ;
* D’assurer la bonne exécution de l’ensemble des fonctionnalités du système.

1. **Besoins non fonctionnels**

Tous les systèmes d’information à un certain point dans leur cycle de vie doivent considérer des besoins non-fonctionnels. Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou de conception. En résumé, ce sont des indicateurs de qualité de l’exécution des besoins fonctionnels :

* **Convivialité :** la plateforme doit être facile à utiliser. En effet, les interfaces doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l’utilisateur.
* **Performance :** l’application doit être performante à travers toutes ses fonctionnalités et répondre à toutes les exigences des utilisateurs d’une manière optimale.
* **Portabilité :** l’application doit être portable, c’est-à-dire avoir une capacité à fonctionner dans différents environnements d'exécution.
* **Rapidité de traitement :** en effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d’exécution des traitements soit instantanée.
* **Sécurité :** la plateforme doit assurer un niveau maximum de sécurité pour les informations traitées.
* **Responsabilité :** l’application doit fonctionner sur n’importe quel type de périphérique (ordinateur, smartphone, tablette).
* **Disponibilité / Fiabilité :** l’application doit être accessible à tout moment 24h /24, 7 jours /7 (sauf en période de maintenance).

1. **Méthodologie de gestion de projet utilisée**

La gestion de projet est le mode de réalisation d'un projet, où l'application des techniques de gestion pendant le cycle de vie du projet permet d'atteindre des objectifs précis.[[5](https://ics.utc.fr/portail_linios/Linios/LINIO_gpfoad/co/intro_6.html)]

Il est difficile de faire en sorte qu’une équipe soit organisée, rapide et efficace sans mettre en place une forme de [méthodologie de projet](https://bubbleplan.net/blog/methodes-gestion-projet-comment-choisir/). Celle-ci servira à structurer et rationaliser l’organisation d’un projet ; mais on en trouve de nombreuses formes dont :

1. **Méthodes traditionnelles**

La méthode classique de gestion de projet, également appelée méthode traditionnelle ou méthode prédictive d’un projet fait allusion à l’approche classique de la gestion de projet. Ces méthodes reposent souvent sur des approches séquentielles qui se basent sur un découpage des projets par tâches bien définies (découpage linéaire et séquentiel du cycle du projet), un cadrage et une planification prédictive et enfin sur le peu d'interaction en cours de projet avec le client. Elles supposent que les projets soient relativement simples, prévisibles, et linéaires, pour pouvoir établir une planification détaillée et la suivre dans le temps. Parmi ces méthodes, celles qui reviennent le plus souvent sont : la méthode Waterfall (dite “en Cascade”) et le cycle en V. Elles opèrent avec de petites équipes, et dans lesquelles les membres des équipes ne dépendent pas les uns des autres pour faire avancer le projet. Si la communication au sein de l’équipe peut se limiter au strict minimum, la gestion traditionnelle est efficace.

Cependant, perçue comme rigide et limitant la créativité, cette approche traditionnelle a été remise en cause par de nouvelles méthodes dites « agiles ».

1. **Méthodes agiles**

Les méthodes agiles constituent un ensemble de pratiques inhérentes à la gestion de projets. Il s’agit d’une méthode de gestion de projet révolutionnaire caractérisée par une démarche itérative et incrémentale, conçue entre autres pour le développement informatique et des projets qui n’ont pas besoin d’un contrôle intensif sur les livrables. En vue de donner pleine satisfaction aux clients, l'objectif principal est d’introduire la flexibilité au sein des organisations. Elles sont légères et adaptables au changement puisque les clients peuvent évaluer la qualité du projet en cours de réalisation et intervenir pour rectifier le tir au besoin. Parmi ces méthodes, celles principalement utilisées sont : la méthode SCRUM, EXtreme Programming (XP), Dynamic systems development method (DSDM), etc.

Si vous travaillez avec une équipe auto-motivée et qui communique en temps réel, ce type de gestion de projet fonctionne bien parce que les membres de l'équipe peuvent rapidement faire des ajustements en cas de nécessité afin de trouver des solutions rapides aux problèmes présents.

1. **Justification de notre choix**

Les méthodes traditionnelles et les méthodes agiles précédemment citées présentent des aspects intéressants mais se distinguent grandement comme vous l’aurez compris, voici d’ailleurs un tableau récapitulant ainsi qu'une image qui nous montrent les principales différences entre une méthode classique de gestion de projet et la méthode SCRUM qui est la méthode agile la plus utilisée de nos jours.

|  |  |  |
| --- | --- | --- |
| **Critères de référence** | **Méthode classique de gestion de projet** | **Méthode Scrum** |
| **Management et organisation** | Organisation hiérarchisée, les ressources sont spécialisées | Organisation plus horizontale et participative |
| **Taille de l’équipe** | Pas de limite de taille | Taille d’équipe limitée car la méthode implique l’organisation de nombreuses réunions, les [**cérémonies Scrum**](https://www.appvizer.fr/magazine/operations/gestion-de-projet/ceremonies-scrum) |
| **Cycle de vie** | Découpage séquentiel et linéaire | Fonctionnement itératif et incrémental |
| **Planification** | Prédictive, tout est cadré en amont | Adaptative, le projet évolue au fil des sprints |
| **Flexibilité** | Ne permet pas de retour en arrière | Méthode flexible qui s’adapte aux évolutions du besoin client |
| **Contrôle qualité** | Le client découvre et valide le produit en fin de projet | Le client intervient plus fréquemment, visualise et valide le projet à différentes étapes |
| **Documentation** | Produite en quantité : elle cadre le projet et sert de support de validation et de contractualisation | Moins fournie, grâce à l’approche incrémentale et les retours réguliers du client |
| **Mesure du succès** | Le respect des engagements prévus initialement (coûts, délais et qualité) | La satisfaction du client et la valeur ajoutée du produit |

Tableau 19: Différence entre Méthodes traditionnelles et Scrum

Au vu de ces éléments, il apparaît clairement que la méthode la plus appropriée dans le cadre de ce projet est la méthode agile SCRUM dont nous allons faire une brève présentation.

* ***Définition :***

Scrum est une méthode de développement agile orientée projet informatique dont les ressources sont régulièrement actualisées. La méthode Scrum tire son nom du monde du rugby, scrum = mêlée. Le principe de base étant d'être toujours prêt à réorienter le projet au fil de son avancement. Elle privilégie la livraison rapide d'un prototype, opérationnel par définition, afin que les clients, donneurs d'ordre et membres de l'équipe puissent l'évaluer.

* ***Rôles :***

Scrum définit trois rôles particuliers, chacun ayant une fonction bien précise, sans aucun rapport hiérarchique entre eux :

* Le Product Owner qui porte la vision du produit à réaliser et travaille en interaction avec l’équipe de réalisation et le client. Il s’agit généralement d’un expert du domaine métier du projet ;
* L'Équipe de développement ou de réalisation qui est en charge de transformer les besoins du client en produit ou service ;
* Le Scrum Master est le garant de la bonne application de la méthode Scrum. Il a un rôle de coach à la fois auprès du Product Owner et auprès de l’équipe.
* ***Fonctionnement :***

Pour bien comprendre la méthodologie Scrum, il est essentiel de savoir faire la différence entre product backlog, sprint backlog et stories :

* La méthode Scrum impose au product-owner en amont de définir le product backlog. Le backlog est élaboré avant le lancement des sprints, dans la phase de préparation (ou sprint 0). Il est utilisé pour planifier l’ensemble des exigences du client et des fonctionnalités du produit ou du service. Chaque besoin est classé par ordre de priorité en fonction de leur valeur ajoutée (pour un produit ou logiciel) ou de leur chronologie (pour un service) ;
* Ensuite, à chaque sprint, lors de la réunion de planification, l’équipe et le product-owner décident ensemble du sprint backlog, le sous-ensemble qui sera réalisé lors du sprint. Chaque sprint est une boucle qui suit la même chronologie : planification (pour estimer et répartir les tâches), mêlée quotidienne (pour synchroniser le travail d’équipe), revue de sprint (pour présenter l’avancement au client), rétrospective (pour établir un plan d’actions d’amélioration) ;
* Chaque sous-ensemble est divisé en stories. Pour chaque stories, l’équipe détaille les sous-tâches à réaliser et y attribue une personne en charge de la réaliser.
* ***Les outils :***

Différents outils de suivi et de calcul d’indicateurs peuvent être utilisés avec la méthode Scrum pour contrôler le déroulement du projet.

Nous vous présentons ci-dessus les principaux :

* Le [Burndown Chart](https://www.nutcache.com/fr/blog/scrum-burndown-chart/) **:** il s’agit d’un graphique simple permettant de visualiser le degré d’avancement de chacune des tâches. Il permet de fournir à l’ensemble de l’équipe une vision claire et actualisée de l’état d’avancement des travaux et de la quantité de travail restante. Il est généralement mis à jour lors de la réunion quotidienne ;
* Le task board : outil central du sprint scrum, ce tableau de bord projet permet de suivre en temps réel la progression de la réalisation des différentes tâches. Il est composé de trois colonnes comportant les tâches à faire, les tâches en cours et les tâches terminées. Généralement un tableau accroché au mur ou affiché sur un grand écran actualisé directement par les développeurs dès que l’activité sur laquelle ils travaillent évolue ;
* La learning matrix: cette « matrice d’apprentissage » a pour objectif d’aider les membres de l’équipe projet à identifier les points positifs du projet et ceux qui nécessitent une amélioration. Utilisée notamment lors de la rétrospective scrum agile, elle fait partie du processus d’amélioration continue du projet. La matrice est divisée en quatre quadrants : ce qui ne va pas et doit être changé, ce qui va bien et qui doit être répété, les nouvelles idées à essayer et les personnes appréciées. Chaque participant peut coller un post-it avec leurs idées dans les différents quadrants, puis chacun peut ensuite voter pour les suggestions proposées. Cela permet au final de dégager des priorités en termes d’organisation notamment pour le sprint suivant et d’améliorer les conditions de travail et la productivité de l’équipe.
* Le storyboard : il s’agit d’un outil graphique permettant d’illustrer les différentes étapes permettant d’atteindre un objectif. Le storyboard ressemble à une bande dessinée, chaque case représentant une étape, et donne un exemple de ce qui est attendu. Cet outil est destiné à la description de l’aspect fonctionnel et non à la conception ou à la démonstration de l’interface graphique.
* Le [graphique Sunset](https://www.nutcache.com/fr/blog/graphique-sunset/) : ce graphique permet de suivre la progression de l’équipe ainsi que l’avancement du projet plus généralement. Il fournit une vue globale du contenu du backlog produit avec un classement par importance des fonctionnalités attendues. Ainsi, la vélocité de l’équipe apparaît clairement et il est possible d’anticiper d’éventuelles modifications des dates de livraison. Il est également possible de suivre la consommation du budget et de prévenir les risques de dérapage. Le graphique Sunset est mis à jour à la fin de chaque scrum sprint.

  Pour clore nous diront que l’approche SCRUM est une méthode agile consacrée à la gestion de projet. Son objectif phare est d’améliorer la productivité des équipes, tout en permettant une optimisation du produit grâce à des feedbacks réguliers du marché. En parallèle, son fonctionnement a pour finalité d’avoir une vue d’ensemble du projet pour chacune des parties prenantes. Et elle permet aussi la réduction des bugs et une mise à jour régulière des priorités.

1. **Méthode et formalisme d’analyse et conception utilisés**
   1. **La méthode MERISE**

MERISE (**M**éthode d’**E**tude et **R**éalisation **I**nformatique pour les **S**ystèmes d’**E**ntreprise)

est une méthodologie de modélisation à usage général dans le domaine du développement de systèmes d’information, du génie logiciel et de la gestion de projet. Introduit pour la première fois au début des années 1980, Merise procède un traitement séparé des données et des processus, où la vue des données est modélisée en trois étapes:

* la conception
* la logique
* la physique

De même, la vue axée sur les processus passe par les trois étapes

* conceptuelle,
* organisationnelle
* opérationnelle.

Ces étapes du processus de modélisation sont parallèles aux étapes du cycle de vie:

* planification stratégique
* étude préliminaire
* étude détaillée
* développement
* mise en œuvre et
* maintenance.

C'est une méthode de conception et de développement de systèmes d’information globale se basant sur les trois sous-systèmes qui sont :

* Le sous-système de pilotage (SSP) ;
* Le sous-système d’informations (SSI) ;
* Et le sous-système opérant (SSO) ;

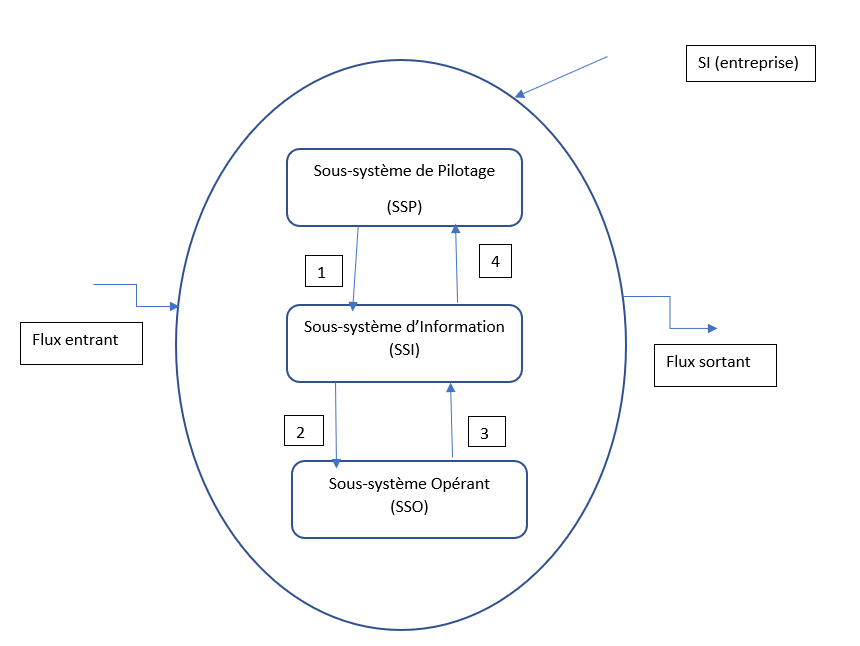


Figure 9: : Schéma systématique de l’entreprise selon Jean Louis Le Moigne

**Légende :**

1 : Décision

2 : Ordre

3 : Résultat de l'exécution de processus

4 : Compte rendu sur le comportement du SSO

1. **Le langage UML**

UML (langage de modélisation unifié) est issu de la fusion des méthodes Booch, OMT (Object Modelling Technique) et OOSE (Object Oriented Software Engineering) ; C’est un langage de modélisation orienté objet c’est-à-dire que toutes les entités modélisées sont des objets ou se rapporte à des objets indépendamment de tout langage de programmation. Il sert également à spécifier, visualiser, concevoir et documenter tous les aspects d'un système d'information. Il est devenu le standard de l’industrie pour toutes les méthodes orientées objets.

De manière courte, UML est une sorte de boîte à outils, qui permet d'améliorer progressivement vos méthodes de travail, tout en préservant vos modes de fonctionnement. Cependant, dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche :

* itérative et incrémentale,
* guidée par les besoins des utilisateurs du système,
* centrée sur l'architecture logicielle.

1. **Justification de notre choix**

Merise et UML sont deux grands principes de « traduction » ou modélisation d'un système d'information. Néanmoins, ils ne sont pas aussi proches qu'on pourrait le penser.

Le choix de l'un ou de l'autre se fait selon trois axes à savoir l'accessibilité, la précision et l'exploitabilité :

* Pour le premier axe (accessibilité) MERISE présente l'intérêt d'avoir des modèles logiques moins détaillés facilement compréhensibles. Tandis qu’UML conçu pour s'adapter à n'importe quel langage de programmation orientée objet (POO), présente plusieurs modèles (diagrammes) dont leurs compréhensions nécessitent une grande attention ;
* En ce qui concerne le deuxième critère (précision), MERISE est moins préférable. Malgré sa clarté, il manque une précision du fait qu'elle est éloignée du langage donc difficile à implémenter alors qu’UML intègre les éléments communs des différents langages, sa volonté est d'être fidèle à la réalisation finale. Elle est beaucoup plus complète avec ses différents diagrammes ;
* Pour en finir avec l'exploitabilité, MERISE est une méthode plus généraliste. Elle donne une vue globale de la solution sans pour autant entrer dans les petits détails. Contrairement à UML qui est conçu pour l'implémentation objet avec ses différents détails et sa portabilité (s'adapte à n'importe quelle plateforme) elle est donc plus exploitable.

Au vu de ses éléments UML se trouve être le meilleur choix dans le cadre de notre application grâce à son adaptabilité aux différents langages de programmation et à l’ensemble de ses diagrammes qui permettent une analyse facile et fidèle à la réalisation finale. Aussi, elle apporte une compréhension rapide du programme à d’autres développeurs externes en cas de reprise du logiciel et facilite également sa maintenance.

**Diagramme de cas d’utilisation**

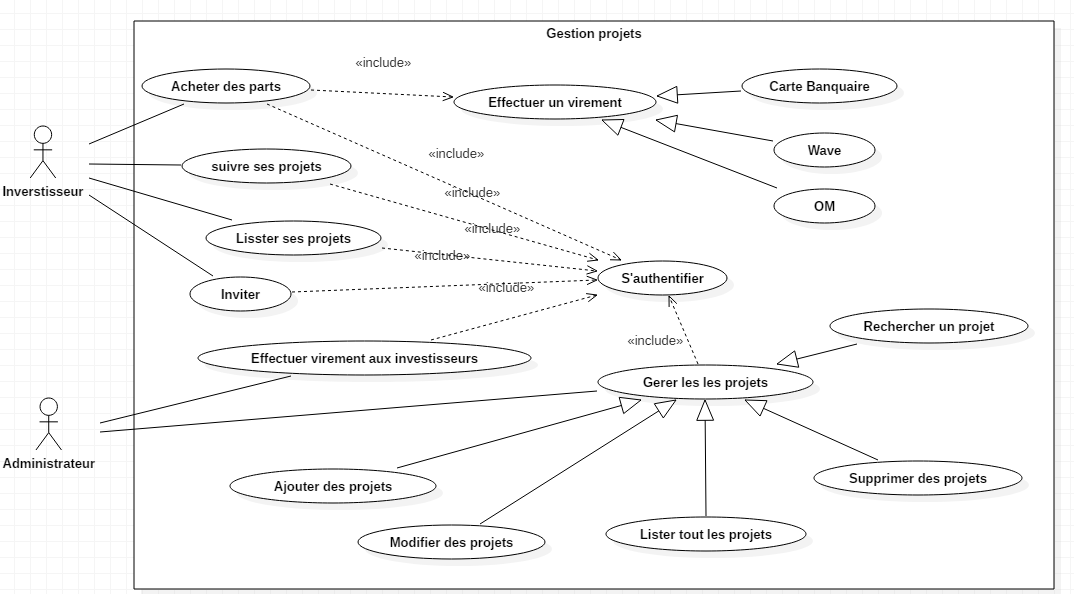
******

Figure 15: Diagramme de cas d'utilisation-Gestion projet

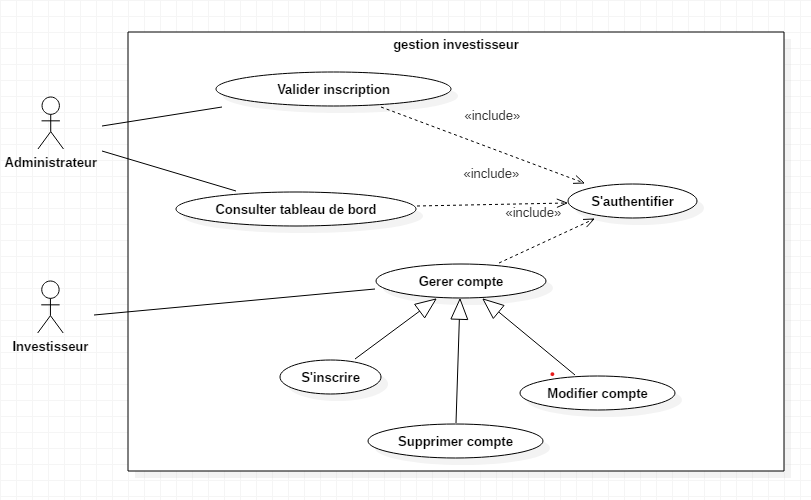
******

Figure 16:  Diagramme de cas d'utilisation-Gestion investisseur

1. **Diagrammes de séquence**

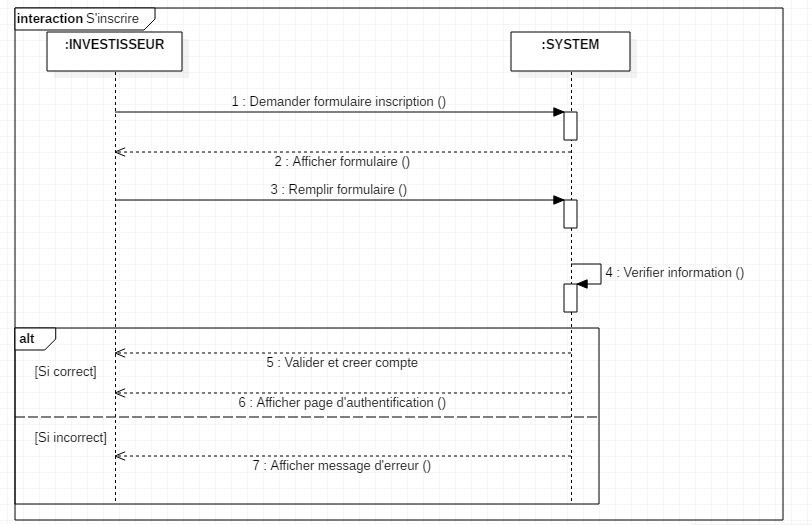


Figure 17: Diagramme de séquence-s’inscrire

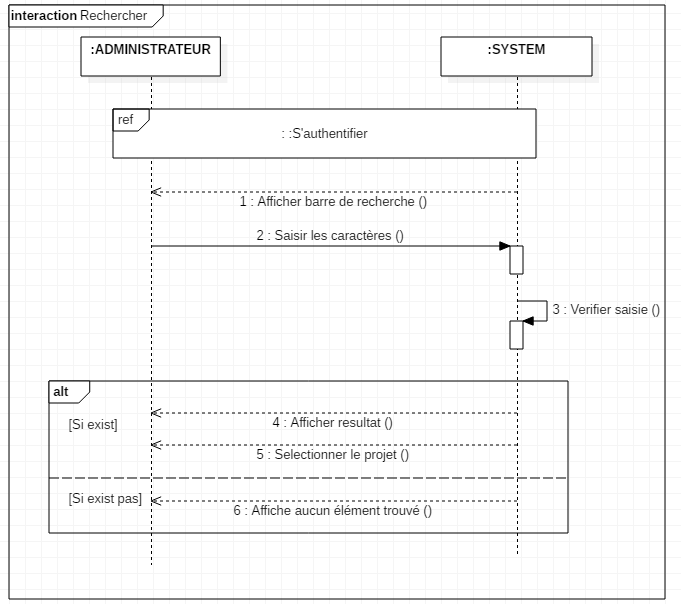


Figure 18: Diagramme de séquence-Rechercher un projet

1. **Diagramme d’activités**

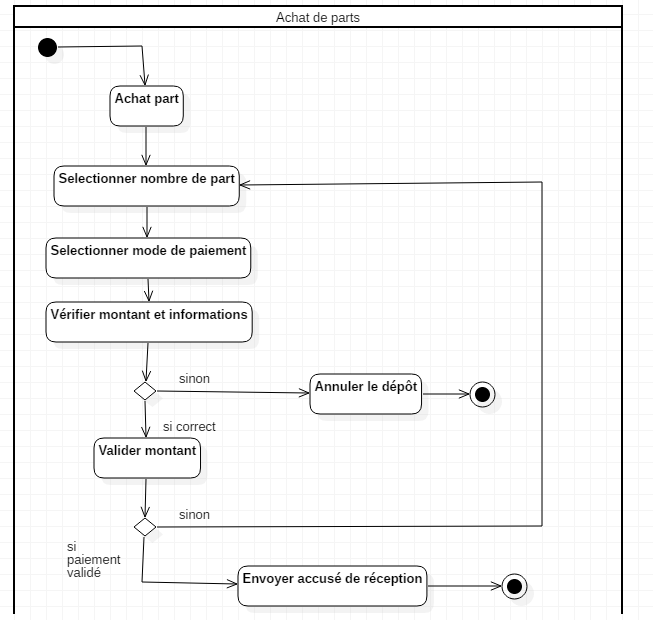


Figure 19:Diagramme d'activités-Achat de parts

1. **Diagramme de classes**

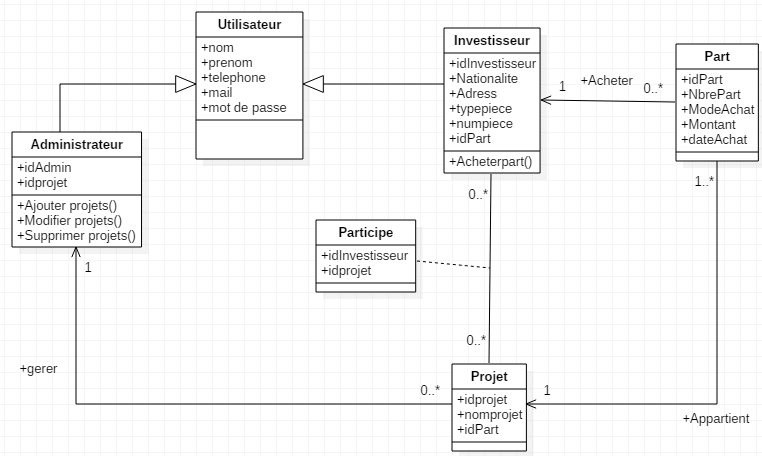
******

Figure 21: Diagramme de classes