

Les diagrammes fonctionnels

Table des matières

I. Contexte	3
II. Connaître les principes de modélisation fonctionnelle (cas d'utilisation et paquets)	3
A. Diagrammes de cas d'utilisation	3
B. Application pratique	4
C. Diagramme de paquetages	8
D. Application pratique	8
III. Connaître les principes de modélisation fonctionnelle (activités and séquence)	9
A. Diagrammes d'activités	9
B. Application pratique	11
C. Diagramme de séquence	12
D. Application pratique	13
IV. Essentiel	14
V. Auto-évaluation	14
A. Exercice	14
B. Test	14
Solutions des exercices	15

I. Contexte

Durée : 1 h

Environnement de travail : Draw.io ou Visual Paradigm

Pré-requis : aucun

Contexte

Avez-vous déjà essayé d'expliquer l'une de vos idées de logiciel à vos amis ? Pas facile... On se rend vite compte qu'il y a souvent un écart conséquent entre notre idée et la façon dont elle est comprise par l'autre. Alors, si on lui demande de coder le logiciel pour nous, sur la simple base de nos explications, on risque de récupérer un produit bien loin de notre idée d'origine.

C'est précisément pour fluidifier la communication entre les développeurs et les personnes concevant l'application que les diagrammes ont été inventés. On peut en distinguer deux grandes catégories : les diagrammes de structure, et les diagrammes de comportements. Grâce à ces diagrammes, nous avons une trace écrite et claire des besoins fonctionnels et techniques de l'application.

Ces diagrammes vont servir de base au chef de projet et au développeur pour créer l'application, en répondant correctement aux besoins utilisateurs.

II. Connaître les principes de modélisation fonctionnelle (cas d'utilisation et paquets)

A. Diagrammes de cas d'utilisation

Définition

Les diagrammes de cas d'utilisation sont une représentation visuelle des fonctionnalités d'un système telles qu'elles sont perçues par un utilisateur. Ils permettent de comprendre facilement le système en mettant l'accent sur l'interaction entre le système et les utilisateurs finaux.

Méthode

Les éléments clés dans un diagramme de cas d'utilisation comprennent :

- **Les acteurs** : ce sont les entités (humains ou systèmes) qui interagissent avec le système. Ils peuvent être des utilisateurs, des systèmes externes ou d'autres systèmes.
- **Les cas d'utilisation** : ils représentent des actions spécifiques que le système peut effectuer en réponse à un acteur. Par exemple, « *Commander un produit* » pourrait être un cas d'utilisation pour un système de commerce électronique.
- **Les relations** : il existe plusieurs types de relations dans un diagramme de cas d'utilisation, notamment les associations (un lien entre un acteur et un cas d'utilisation), les généralisations (un lien d'héritage entre deux cas d'utilisation ou deux acteurs), et les inclusions et extensions (des liens qui décrivent les dépendances entre les cas d'utilisation).

Pour dessiner un diagramme de cas d'utilisation, vous pouvez suivre les étapes suivantes :

1. Identifiez les acteurs.
2. Identifiez les cas d'utilisation.
3. Dessinez les relations.

B. Application pratique

Méthode	Trouver les acteurs
---------	---------------------

Réalisons un diagramme de cas d'utilisation pour une application de commande de nourriture en ligne. Rappelons que les trois étapes de création d'un diagramme sont : l'identification des acteurs, l'identification des actions, et l'identification des relations.

Les acteurs représentent les rôles des entités qui interagissent avec le système. Ces acteurs peuvent être des personnes, des organisations, des composants logiciels ou matériels externes, ou même d'autres systèmes. En d'autres termes, tout ce qui « agit » sur le système ou en interagit peut être considéré comme un acteur.

Voici 2 étapes vous permettant de dresser une liste exhaustive des acteurs de votre application :

1. **Identifier les utilisateurs** : cette première étape consiste à identifier tous les utilisateurs potentiels du système. Cela peut inclure les utilisateurs directs, comme les clients ou les employés, mais aussi les utilisateurs indirects, comme les administrateurs ou le personnel d'assistance. Nous pouvons à ce moment-là identifier ces utilisateurs par leurs rôles.
2. **Identifier les systèmes externes** : les systèmes externes qui interagissent avec votre système peuvent également être des acteurs. Par exemple, si votre application utilise un service de paiement externe, ce service serait un acteur dans votre diagramme de cas d'utilisation.

Exemple Identification des acteurs

Dans notre cas, nous pourrions définir beaucoup d'acteurs.

Mais, pour simplifier les choses, nous allons en choisir uniquement deux : le client et le restaurateur.



Client



Restaurateur

Méthode Trouver les cas d'utilisation

Les cas d'utilisation sont des séquences spécifiques d'actions ou d'événements qu'un système peut effectuer en interaction avec un acteur. Pour définir les cas d'utilisation, vous pouvez suivre ces étapes :

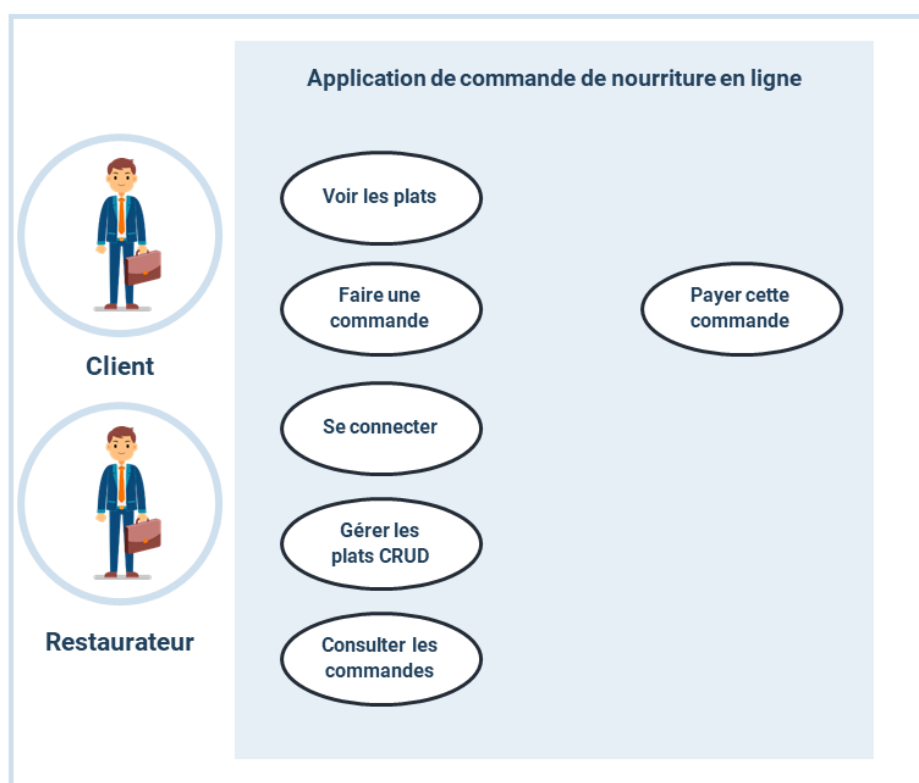
1. **Identifier les fonctionnalités du système** : commencez par dresser la liste de toutes les fonctionnalités ou tâches que votre système est censé accomplir. Par exemple, pour un site de commerce électronique, cela pourrait inclure des tâches comme « *rechercher un produit* », « *ajouter un produit au panier* » ou « *passer une commande* ».
2. Depuis ces fonctionnalités, **réfléchir si d'autres fonctionnalités ou suite d'actions en découlent**. Avoir une liste des fonctionnalités principales nous aide souvent à définir les autres fonctionnalités plus secondaires, ou qu'on aurait oubliées.
3. **Renommer chaque cas d'utilisation si nécessaire** : chaque cas d'utilisation doit avoir un nom clair et descriptif qui reflète l'interaction qu'il représente. Par exemple, au lieu de nommer un cas d'utilisation « *Commande* », vous pourriez le nommer « *Passer une commande* » pour être plus précis. En cherchant à renommer nos cas d'utilisation, on en trouve des nouveaux. « *Commande* » pourrait devenir « *Passer une commande* », mais aussi « *Annuler une commande* », « *Consulter une commande* », etc.

Exemple Identification des cas d'utilisation

Dans une application de commande de nourriture en ligne, nous devons pouvoir voir les plats proposés, faire une commande, payer cette commande.

Nous devons aussi pouvoir gérer les produits ou plats proposés dans l'application. C'est-à-dire qu'il nous faudra pouvoir créer, lire, modifier ou supprimer un plat présent dans l'application. C'est ce qu'on appelle un CRUD (*create, read, update, delete*).

Nous devons aussi créer un module de connexion.



Méthode Trouver les relations

Il existe différents types de relations dans un diagramme de cas d'utilisation :

- **L'association** : c'est le type de relation le plus courant dans un diagramme de cas d'utilisation. Une association est simplement une ligne reliant un acteur à un cas d'utilisation pour indiquer que l'acteur participe à ce cas d'utilisation. Par exemple, si un acteur « *Client* » peut « *Passer une commande* », vous dessinerez une ligne d'association entre l'acteur « *Client* » et le cas d'utilisation « *Passer une commande* ».
- **L'inclusion** : une inclusion est une relation dans laquelle un cas d'utilisation « *inclut* » un autre cas d'utilisation, ce qui signifie que la réalisation du premier cas d'utilisation nécessite obligatoirement l'exécution du second. Par exemple, dans un système de commerce électronique, le cas d'utilisation « *Passer une commande* » pourrait inclure le cas d'utilisation « *Vérifier la disponibilité du produit* ». L'inclusion est représentée par une flèche pointillée avec le mot-clé « *<<include>>* ».
- **L'extension** : c'est une relation dans laquelle un cas d'utilisation peut étendre un autre cas d'utilisation par des comportements supplémentaires, qui sont généralement optionnels. Par exemple, le cas d'utilisation « *Passer une commande* » pourrait être étendu par le cas d'utilisation « *Appliquer un code de réduction* ». L'extension est représentée par une flèche pointillée avec le mot-clé « *<<extend>>* ».
- **La généralisation** : c'est une relation d'héritage entre deux cas d'utilisation ou deux acteurs. Cela signifie qu'un cas d'utilisation ou un acteur « *enfant* » hérite de toutes les propriétés et du comportement d'un cas d'utilisation ou d'un acteur « *parent* ». Par exemple, dans une application bancaire, l'acteur « *Client Premium* » pourrait être une généralisation de l'acteur « *Client* », avec des cas d'utilisation supplémentaires tels que « *Accéder à des services premium* ». La généralisation est représentée par une ligne solide avec une flèche creuse.

Exemple Les relations

Maintenant, nous devons définir les relations entre les différents acteurs et les différentes actions.

Pour commencer, nous pouvons nous demander qui sont les acteurs des différentes actions. Après réflexion, nous pourrions arriver à cette liste :

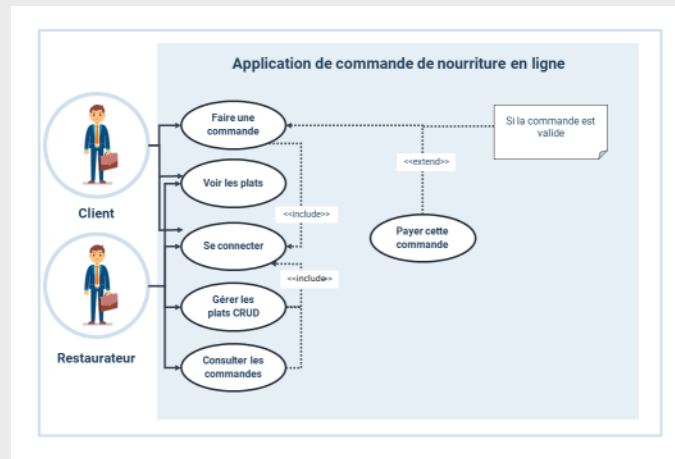
- Faire une commande : Client
- Payer cette commande : Client
- Voir les plats : Client, Restaurateur
- Se connecter : Client, Restaurateur
- Gérer les plats CRUD : Restaurateur
- Consulter les commandes : Restaurateur

Maintenant que nous avons les liens entre les acteurs et les actions, nous pouvons nous poser la question des liens entre les différentes actions.

Une action est-elle possible uniquement si une autre action a été réalisée avant ? Si oui, nous pouvons ajouter un lien « *include* ».

Une action est-elle la suite d'une autre action, avec une condition ? Si oui, nous pouvons ajouter un lien « *extend* ».

En ayant répondu à ces questions, nous obtenons le résultat final suivant :



C. Diagramme de paquetages

Définition Diagramme de paquetages

Un diagramme de paquetages est un type de diagramme structurel UML qui montre comment les éléments d'un système sont regroupés en paquets et comment les paquets sont dépendants les uns des autres. Il permet d'avoir une vision de l'application plus globale. Il peut être utilisé dans de nombreuses situations, mais on peut en distinguer deux principales : regrouper les cas d'utilisation en différentes parties pour avoir une vue d'ensemble du projet, moins détaillée, et construire différentes classes à partir du diagramme de classes. On pourrait dire que ce diagramme est un résumé d'un autre diagramme.

Les éléments clés dans un diagramme de paquetages comprennent :

- **Les paquets** : il s'agit de conteneurs logiques qui regroupent des éléments connexes du système.
- **Les classes** : elles sont incluses dans chaque paquet (on peut choisir de les détailler ou non).
- **Les interfaces** : elles définissent les contrats entre les classes et les autres éléments du système (on peut choisir de les détailler ou non).
- **Les dépendances** : il s'agit de relations qui montrent comment les différents paquets dépendent les uns des autres.

Méthode

Pour dessiner un diagramme de paquetages, vous pouvez suivre les étapes suivantes :

1. Identifiez les paquets.
2. Identifiez les classes et les interfaces dans chaque paquet.
3. Dessinez les dépendances entre les paquets.

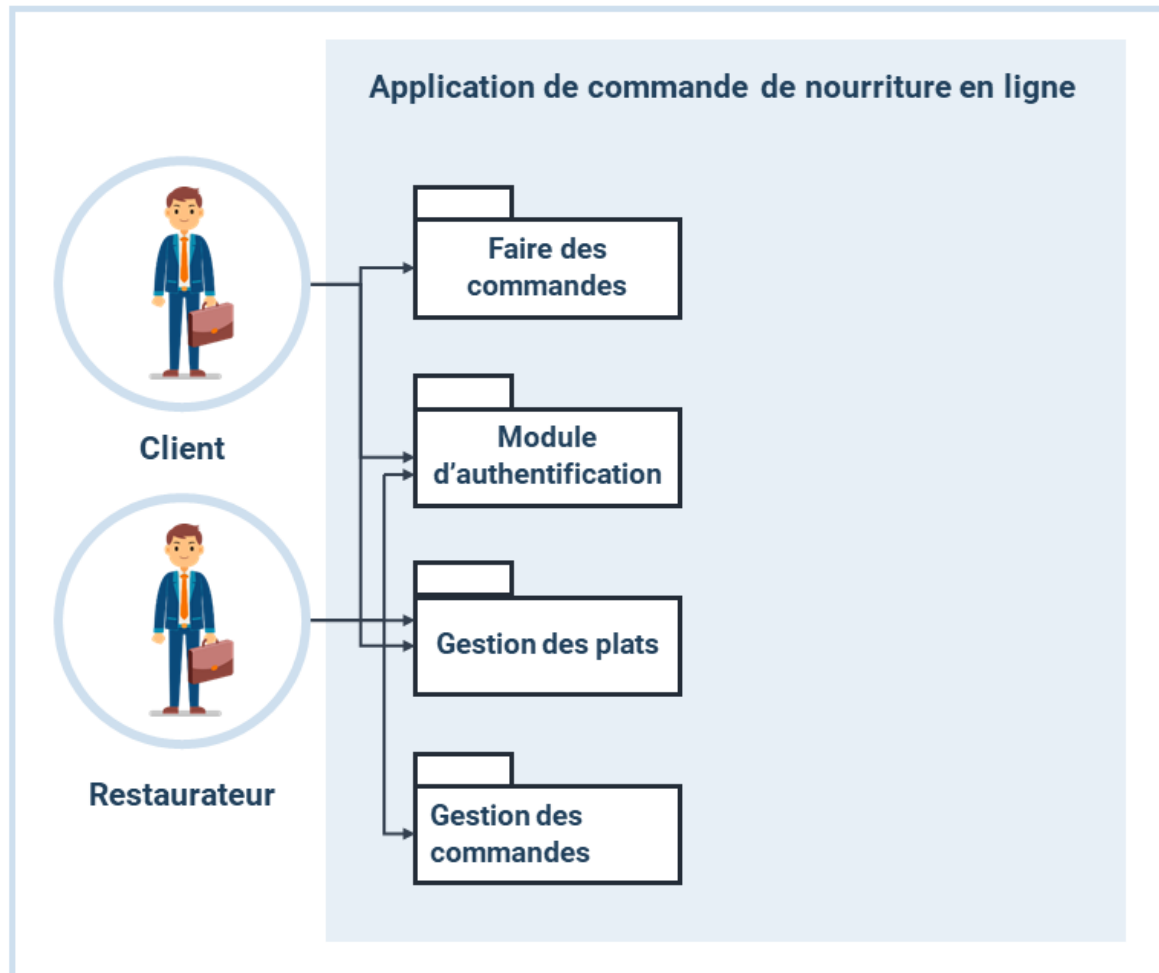
D. Application pratique

Exemple

Nous pouvons reprendre notre exemple d'application pour commander de la nourriture. Notre but est de transformer notre diagramme de cas d'utilisation en diagramme de paquetages. Nous n'allons donc pas gérer les classes, notre objectif est uniquement de regrouper certaines actions.

En soit, dans notre application, nous pouvons définir différents univers :

- La gestion des plats pour un admin
- La réalisation des commandes pour un client
- La gestion des commandes



III. Connaître les principes de modélisation fonctionnelle (activités and séquence)

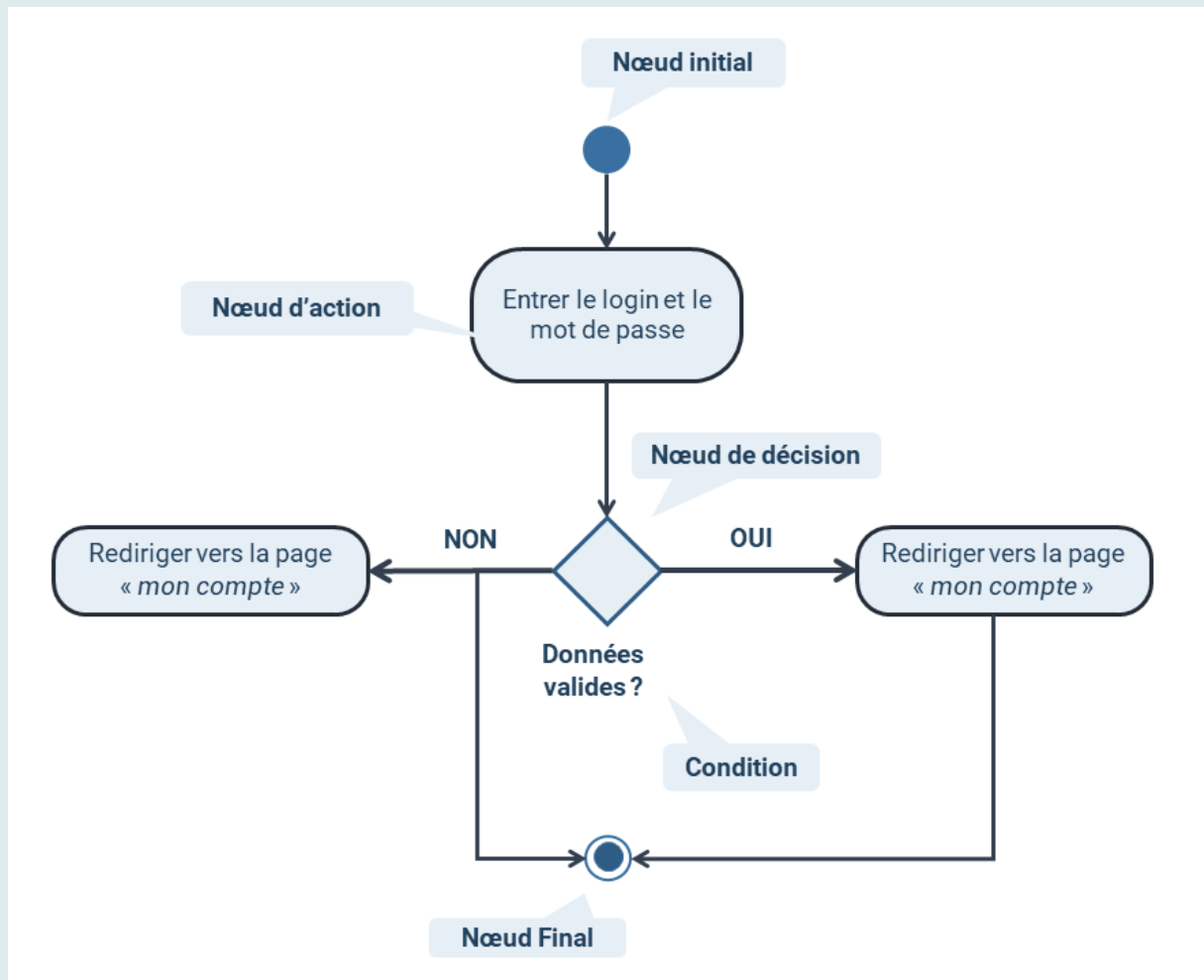
A. Diagrammes d'activités

Définition

Les diagrammes d'activités sont des diagrammes de flux de travail qui illustrent les étapes d'un processus de bout en bout, y compris les points de décision, les chemins parallèles, le début et la fin. Ils sont particulièrement utiles pour visualiser le flux d'opérations dans le contexte de cas d'utilisation.

Les éléments clés d'un diagramme d'activités comprennent :

- Les activités : ce sont des unités de travail qui doivent être accomplies.
- Les transitions : elles représentent le flux de contrôle d'une activité à une autre (les flèches sur le schéma).
- Les décisions : elles représentent des points où le flux de contrôle peut diverger en plusieurs chemins.
- Les signaux : ce sont des événements synchrones qui peuvent influencer le flux de contrôle.



Exemple

Pourquoi faire ce diagramme ?

Ce diagramme est extrêmement utile lorsque nous devons respecter des règles métier qui nous étaient inconnues avant la création de l'application. Par exemple, si nous devons créer une application bancaire. Lors de la création d'un compte pour une personne, nous allons faire face à différentes règles, qui vont limiter nos actions, en fonction de l'âge de la personne, de ses revenus, etc.

En tant que développeur, nous ne sommes pas de ce métier-là. Il est donc fondamental pour nous de noter le processus de création de compte bancaire avec toutes ces règles avant le développement. Ainsi, lors du développement, nous aurons un document expliquant le processus précisément.

B. Application pratique

Exemple

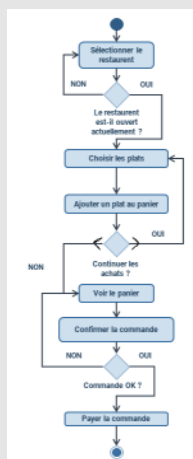
Nous allons faire un diagramme d'activité, sur l'exemple précédent, l'application de commande de nourriture. Nous voulons représenter l'activité de faire une commande sur le site.

Le but est donc de représenter le comportement utilisateur lors de la commande sur notre site. Nous allons donc représenter les choix que pourra faire l'utilisateur, ainsi que les réactions de notre application.

Méthode Comment réaliser ce diagramme

1. **Penser le déroulé des activités** : commencez par énumérer toutes les activités impliquées dans le processus de commande. Le but est de vous mettre à la place de l'utilisateur, d'imaginer chaque interaction qu'il va avoir, et chaque interaction qu'aura la machine elle aussi. Vous pouvez même faire le test sur une vraie application du même type pour vous aider à être exhaustif dans le processus. Par exemple :
 - Sélectionner le restaurant
 - Choisir les plats
 - Ajouter les plats au panier
 - Confirmer la commande
 - Payer pour la commande
2. **Identifier les décisions** : ces points de décision peuvent être des endroits où l'utilisateur doit faire un choix ou où le système doit prendre une décision. Par exemple, après avoir ajouté un plat au panier, l'utilisateur peut décider de passer commande, ou de continuer ses achats.
3. **Créer le diagramme** : commencez par un cercle rempli pour indiquer le point de départ. Puis dessinez des flèches du point de départ vers chaque activité, en fonction de l'ordre dans lequel elles se produisent. Pour chaque activité, utilisez un rectangle arrondi pour la représenter. Les transitions sont représentées par des flèches reliant les activités. Pour chaque décision, utilisez un losange pour la représenter et étiquetez chaque branche sortant du diamant pour indiquer le choix ou la condition qui détermine cette branche. Terminez par un cercle avec une bordure pour indiquer le point d'arrivée.
4. **Vérifier le diagramme** : une fois que vous avez terminé, passez en revue le diagramme pour vous assurer qu'il représente correctement le processus de commande. Assurez-vous que toutes les activités sont présentes, que les transitions sont correctes, et que toutes les décisions sont prises en compte. Grâce à ce diagramme, vous pourrez vous mettre à la place de l'utilisateur, et voir si votre processus est bon ou s'il ne répond pas à la demande.

Voici un exemple de ce qu'on pourrait obtenir :



C. Diagramme de séquence

Définition

Les diagrammes de séquences sont des diagrammes d'interaction UML qui illustrent comment les objets interagissent dans un scénario de cas d'utilisation. Ils montrent les interactions entre les objets organisées par séquence temporelle.

Les éléments clés d'un diagramme de séquences comprennent :

- Les objets : ce sont les entités qui participent à l'interaction.
- Les messages : ce sont les communications entre les objets,
- Les lignes de vie : elles représentent le temps qui passe pour un objet dans le diagramme.
- La durée : elle représente le temps nécessaire pour exécuter une tâche.

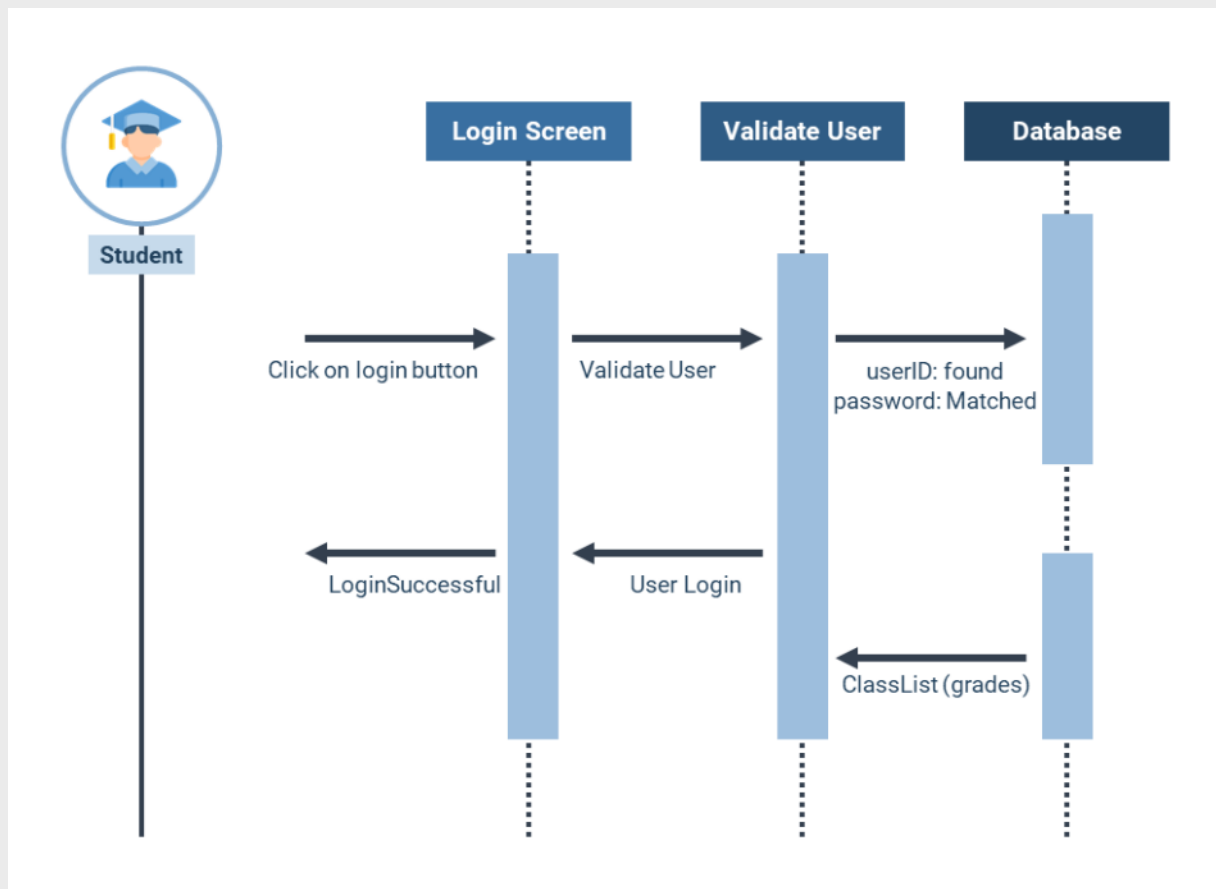
Méthode

Pour dessiner un diagramme de séquences, suivez ces étapes :

1. Identifiez les objets.
2. Identifiez les messages entre les objets.
3. Dessinez les lignes de vie et la durée.

Exemple

Système de connexion d'un étudiant



D. Application pratique

Méthode

Nous allons réaliser le diagramme de séquence de l'application de commande de nourriture en ligne. Ainsi, nous aurons quatre schémas différents pour notre application.

Commençons par le scénario de base : un utilisateur qui souhaite passer une commande depuis son smartphone. Le diagramme de séquence démarre avec l'acteur « *Utilisateur* » qui ouvre l'application et se connecte à son compte. L'application affiche alors une liste de restaurants disponibles.

Pour réaliser notre diagramme de séquence, il va falloir penser au comportement utilisateur, et penser à tous les aller-retours à l'intérieur de notre application que ses actions impliquent.

Exemple

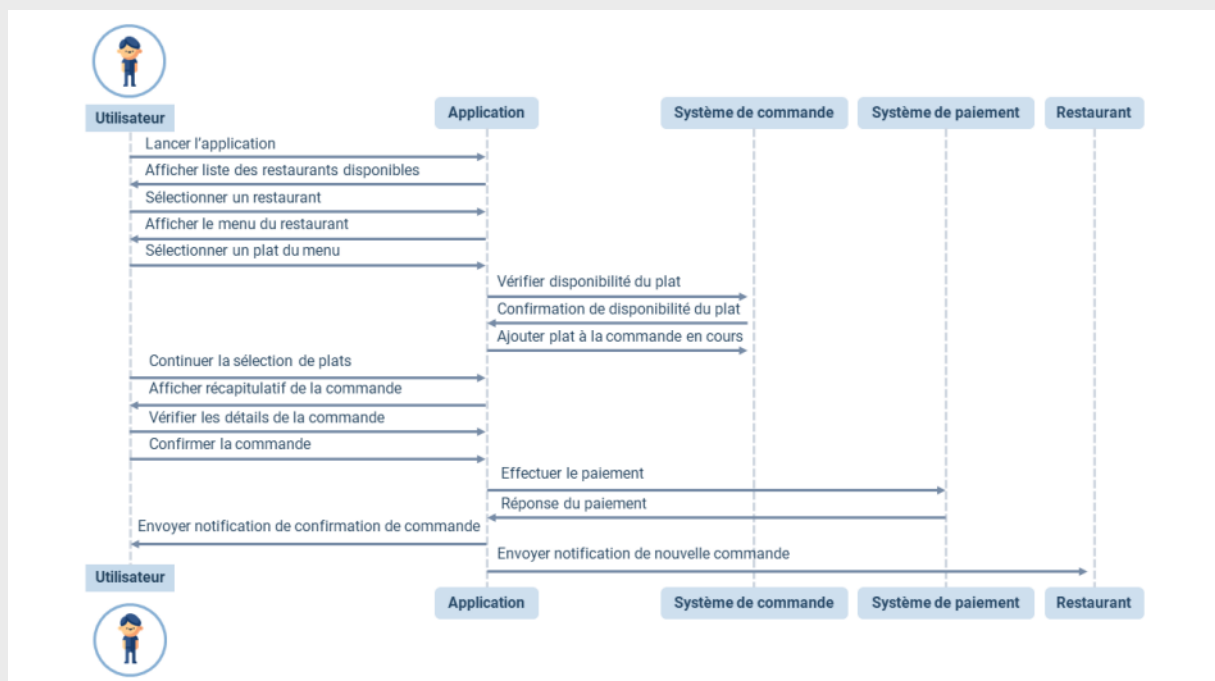
Nous pourrions alors avoir cette réflexion : l'utilisateur sélectionne un restaurant spécifique à partir de la liste et parcourt le menu proposé. Chaque fois qu'il sélectionne un plat, l'application envoie une demande au système de commande pour vérifier la disponibilité de ce plat. Si le plat est disponible, le système de commande l'ajoute à la commande en cours.

Une fois que l'utilisateur a sélectionné tous les plats souhaités, il peut passer à l'étape suivante : la validation de la commande. L'application affiche un récapitulatif de la commande, comprenant les plats sélectionnés, les quantités, les prix et le total. L'utilisateur peut vérifier les détails de la commande et apporter des modifications si nécessaire.

Lorsque l'utilisateur confirme la commande, l'application interagit avec le système de paiement en ligne pour effectuer le paiement. Le système de paiement demande les informations nécessaires à l'utilisateur, telles que les détails de sa carte de crédit. Une fois les informations de paiement fournies, le système de paiement effectue la transaction et renvoie une réponse à l'application, indiquant si le paiement a été accepté ou non.

Si le paiement est accepté, l'application envoie une notification à l'utilisateur, confirmant que sa commande a été passée avec succès. En parallèle, le système de commande envoie une notification au restaurant, leur indiquant qu'une nouvelle commande a été passée.

Le fruit de notre réflexion est ce schéma :



IV. Essentiel

Ce cours vous a permis de maîtriser les principes fondamentaux de la modélisation fonctionnelle à travers différents concepts et diagrammes. Les diagrammes de cas d'utilisation représentent les interactions entre les utilisateurs et un système, en identifiant les acteurs, les cas d'utilisation et en établissant des relations entre eux. Nous avons appris à regrouper ces éléments en diagrammes de paquetages pour une vue d'ensemble organisée. Nous avons aussi vu comment les diagrammes d'activités aident à capturer les flux de travail et les processus, en mettant en avant les actions, transitions et décisions. Nous avons également découvert les diagrammes de séquences, utiles pour visualiser les interactions dynamiques entre les objets dans un scénario donné. Maintenant que vous connaissez ces notions, vous avez de solides bases en modélisation fonctionnelle. Cet outil est fondamental lors de la conception d'une application, phase clé et fondatrice d'un bon développement.

V. Auto-évaluation

A. Exercice

Vous êtes un concepteur de logiciels travaillant pour une start-up qui développe une application mobile de suivi des habitudes. L'application permet aux utilisateurs de suivre différentes habitudes, comme la lecture, l'exercice, la méditation, le sport, etc. Le but est de réussir à accomplir ses objectifs. L'application offre des récompenses sous forme de badges lorsque l'utilisateur atteint certains jalons. Quand l'utilisateur considère qu'une habitude est entrée dans son mode de vie, il peut la marquer comme réalisée. Cela lui permet d'avoir une liste de ses objectifs de vie en cours.

Les utilisateurs peuvent créer un compte, se connecter, ajouter/supprimer des habitudes, marquer une habitude comme réalisée, voir le suivi de leurs habitudes et gagner des badges.

Question 1

[solution n°1 p.17]

En vous basant sur le contexte ci-dessus, dessinez un diagramme de cas d'utilisation qui montre toutes les interactions possibles entre l'utilisateur et l'application de suivi des habitudes. Assurez-vous d'inclure tous les cas d'utilisation pertinents et les acteurs.

Question 2

[solution n°2 p.17]

En vous basant sur le contexte ci-dessus, dessinez un diagramme d'activités qui illustre le processus d'attribution des badges à un utilisateur lorsque celui-ci atteint un certain jalon dans le suivi de ses habitudes.

B. Test

Exercice 1 : Quiz

[solution n°3 p.18]

Question 1

Les diagrammes de cas d'utilisation représentent les fonctionnalités d'un système telles qu'elles sont perçues par un utilisateur.

- ☐ Vrai
- ☐ Faux

Question 2

Les diagrammes de paquetages sont utilisés pour représenter les interactions entre les objets dans un scénario de cas d'utilisation.

- ☐ Vrai
- ☐ Faux

Question 3

Quels sont les éléments clés d'un diagramme d'activités ?

- ☐ Acteurs, cas d'utilisation, relations
- ☐ Activités, transitions, décisions
- ☐ Objets, messages, lignes de vie

Question 4

Quels sont les éléments clés d'un diagramme de séquence ?

- ☐ Acteurs, cas d'utilisation, relations
- ☐ Activités, transitions, décisions
- ☐ Objets, messages, lignes de vie

Question 5

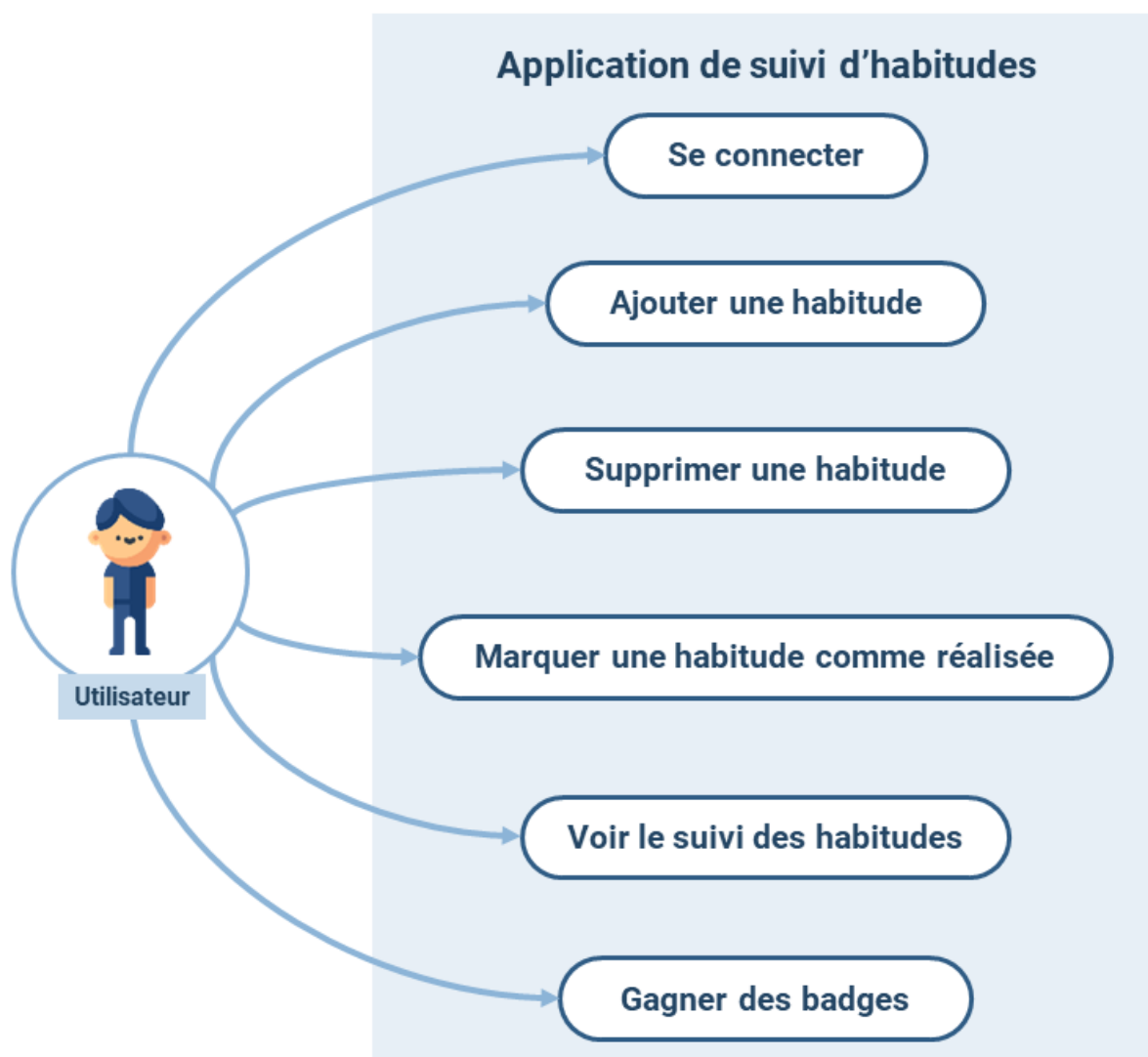
Les diagrammes de séquences montrent les interactions entre les objets organisées par séquence temporelle.

- ☐ Vrai
- ☐ Faux

Solutions des exercices

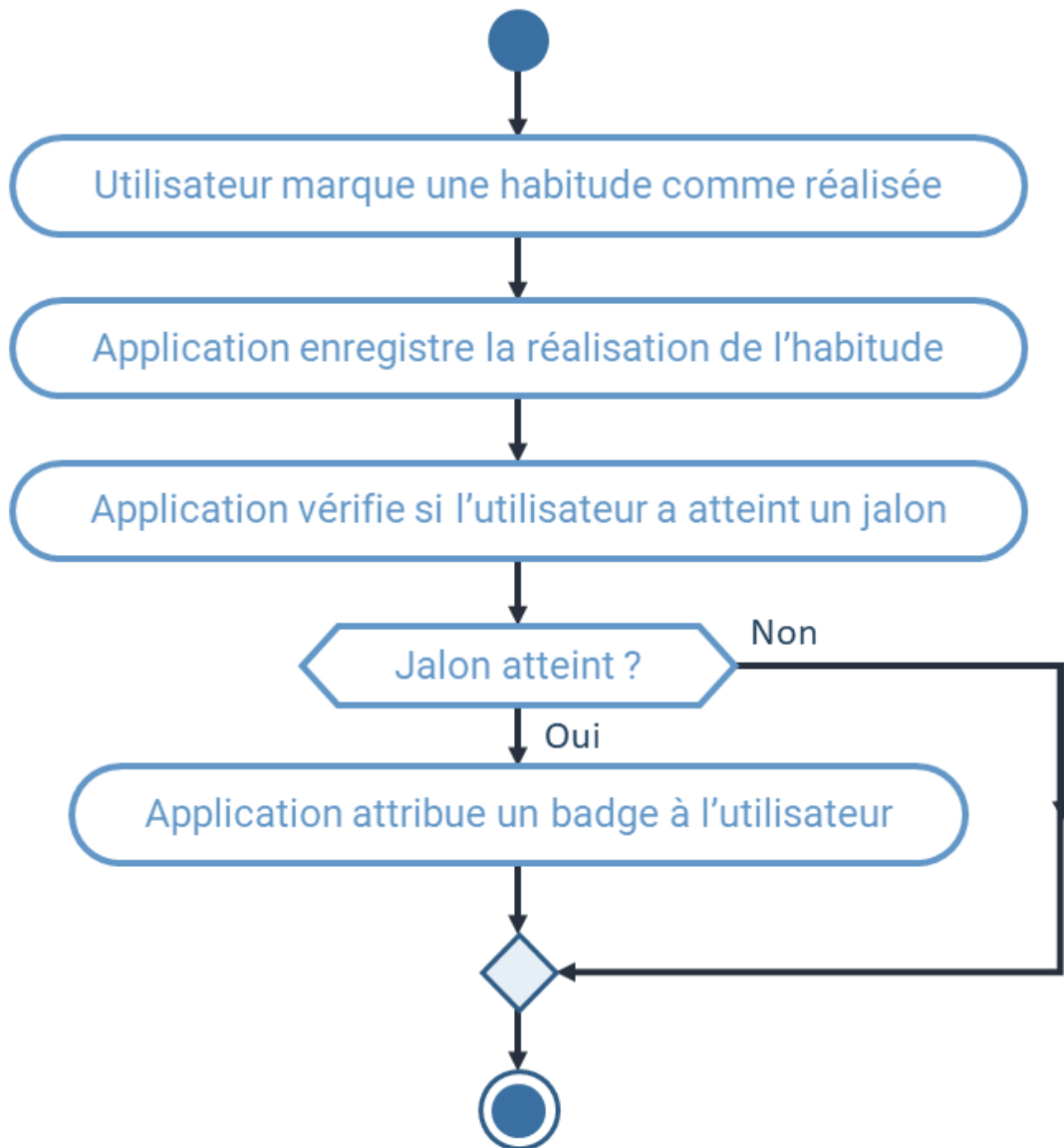
p. 14 Solution n°1

Un diagramme de cas d'utilisation montre les interactions possibles entre l'utilisateur et le système. Dans ce cas, l'utilisateur peut réaliser plusieurs actions comme se connecter, ajouter/supprimer des habitudes, marquer une habitude comme réalisée, voir le suivi de leurs habitudes et gagner des badges. Plusieurs réponses sont possibles, en fonction des fonctionnalités que vous souhaitez gérer dans l'application, et de votre niveau de précision. Le screencast présente une autre manière de répondre à cette question.



p. 14 Solution n°2

Le diagramme d'activités illustre le processus d'attribution des badges à l'utilisateur. L'application vérifie si l'utilisateur a atteint un certain jalon, et si oui, il reçoit un badge. Plusieurs réponses sont possibles, en fonction des fonctionnalités que vous souhaitez gérer dans l'application, et de votre niveau de précision. Le screencast présente une autre manière de répondre à cette question.




Exercice p. 14 Solution n°3

Question 1

Les diagrammes de cas d'utilisation représentent les fonctionnalités d'un système telles qu'elles sont perçues par un utilisateur.

☒ Vrai

☐ Faux


 Les diagrammes de cas d'utilisation sont une représentation visuelle des fonctionnalités d'un système du point de vue de l'utilisateur. Ils mettent l'accent sur l'interaction entre le système et les utilisateurs finaux, permettant ainsi de comprendre facilement le système du point de vue de l'utilisateur.

Question 2

Les diagrammes de paquetages sont utilisés pour représenter les interactions entre les objets dans un scénario de cas d'utilisation.

☐ Vrai

☒ Faux

 Les diagrammes de paquetages sont des diagrammes de structure qui montrent comment les éléments d'un système sont regroupés en paquets et comment les paquets sont dépendants les uns des autres. Ils permettent d'avoir une vision plus globale de l'application en regroupant les éléments connexes du système.


Question 3

Quels sont les éléments clés d'un diagramme d'activités ?

☐ Acteurs, cas d'utilisation, relations

☒ Activités, transitions, décisions

☐ Objets, messages, lignes de vie

 Les éléments clés d'un diagramme d'activités sont les activités, qui représentent les unités de travail à accomplir, les transitions, qui représentent le flux de contrôle d'une activité à une autre, et les décisions, qui représentent les points où le flux de contrôle peut diverger en plusieurs chemins.


Question 4

Quels sont les éléments clés d'un diagramme de séquence ?

☐ Acteurs, cas d'utilisation, relations

☐ Activités, transitions, décisions

☒ Objets, messages, lignes de vie


 Les éléments clés d'un diagramme de séquence sont les objets, qui représentent les entités qui participent à l'interaction, les messages, qui représentent les communications entre les objets, et les lignes de vie, qui représentent le temps qui passe pour un objet dans le diagramme.

Question 5

Les diagrammes de séquences montrent les interactions entre les objets organisées par séquence temporelle.

☒ Vrai

☐ Faux

 Les diagrammes de séquences illustrent la façon dont les objets interagissent dans un scénario de cas d'utilisation, en montrant les interactions entre les objets organisées par séquence temporelle. Ils permettent de visualiser le déroulement des interactions entre les objets dans le temps.