

Cahier des charges

1. Présentation du projet

1.1. En une phrase

L'application que nous allons développer est un programme qui proposera à un groupe d'amis une **soirée** entre eux en fonction de leurs **disponibilités**, de leur **situation** géographique, ainsi que de leurs **préférences**.

1.2. Spécificités

1.2.1. Contexte et besoin

Nous pouvons constater que beaucoup de soirées finissent en brouillon car elles sont soit mal organisées, soit elles ne plaisent pas à tout un groupe d'amis. Une soirée avortée pour ces causes et c'est tout le reste du week-end qui en pâtit ! Se rendre dans des endroits en temps et en heure et qui plaisent à tout un groupe d'amis n'est pas une mince affaire ! C'est pourquoi nous proposons une application permettant d'organiser une soirée entre amis en leur proposant des endroits qui plairont à tout le monde tout en leur garantissant les meilleurs trajets possibles en fonctions de leurs préférences.

1.2.2. Public visé

Les personnes ciblées sont les **adultes** âgés entre **18 et 35 ans** qui aiment passer des soirées entre amis, faisant confiance aux algorithmes et qui habitent à **Paris intra-muros**. Les utilisateurs de cette applications seront des détenteurs de **smartphone** possédant l'accès au réseau internet mobile.

1.2.3. Fonctionnalités principales de l'application

L'application devra proposer au groupe d'amis **trois endroits** définis **chronologiquement** où ils se réuniront à chaque fois :

- un **bar** où prendre l'apéro
- un **restaurant** pour manger
- une **boîte** de nuit

L'organisation d'une soirée s'effectuera sur les critères suivants :

- la **situation géographique** de chacun
- les **horaires de disponibilité** de chaque individu du groupe
- les **goûts** de chacun

- le **budget** général du groupe

Le programme devra définir un trajet pour le déplacement de chaque utilisateur dans le groupe et ceci entre chaque étape.

2. Exigences fonctionnelles

2.1. Définition des différents critères

2.1.1. La situation géographique

L'application prendra en compte la situation géographique de chaque personne dans le groupe afin de choisir des endroits de rencontre qui permettront à chaque fois d'optimiser le temps de trajet, **individuellement** pour les trajets maison/bar et boîte de nuit/maison ainsi que **collectivement** pour les trajets bar/restaurant et restaurant/boîte de nuit. L'application devra donner un itinéraire de transport à chaque utilisateur qui le mènera chez elle à la fin de la soirée grâce à **l'adresse de son domicile** préalablement renseignée. L'application proposera des moyens de transports **uniquement parisiens** et les lieux de rendez-vous seront uniquement dans **Paris intra-muros**. (Détails au point 2.2)

API utilisés :

- Google Maps

2.1.2. Les horaires de disponibilité

Chaque utilisateur aura préalablement renseigné ses horaires de disponibilité sous forme de **plage horaire** avec une heure de début et une heure de fin. L'application prendra en compte les horaires de début de chacun afin de coordonner le regroupement des individus du groupe dans le premier endroit (bar) avec une liberté de **30 minutes** entre le premier arrivant et le dernier. L'heure de fin de disponibilité de chacun devra être prise en compte afin de proposer un itinéraire **retour** en fonction des horaires de fonctionnement des différents transports. (Détails au point 2.2)

2.1.3. Les goûts de chacun

Chaque utilisateur aura préalablement renseigné ses goûts afin que l'application puisse proposer au mieux un restaurant qui plaira à toutes les personnes du groupe. Il y aura pour chaque utilisateur ce qu'il **aime** et ce qu'il **n'aime pas**.. (Détails au point 2.2)

Voici la liste des goûts culinaires que l'utilisateur pourra sélectionner pour le restaurant :

- Fast-food
- Thaï
- Chinois
- Japonais
- Indien
- Africain
- Oriental
- Américain
- Latin
- Français
- Italien

De plus, l'organisateur pourra choisir un thème pour le bar de la soirée parmi la liste suivante :

- Bar à vins
- Bar à bières
- Bar à chicha

API utilisés :

- Google Maps

2.1.4. Les moyens de transport

Chaque utilisateur aura pour les trajets **individuels** (trajet pour aller au bar et trajet pour rentrer de boîte) le choix entre **trois** itinéraires. Un en **transports en commun**, un à **vélo** et un en **voiture**. Lors des trajets en groupe (trajet bar/restaurant et trajet restaurant/boîte), ce sera à **l'organisateur** de la soirée de choisir le moyen de transport que le groupe utilisera et ceci à chaque trajet. L'application proposera uniquement des moyens de transports **parisiens**. (Détails au point **2.2**)

- Google Maps
- JCDecaux

2.1.5. Le budget général du groupe

Avant de générer une soirée, l'application demandera à chaque utilisateur le **budget** qu'il peut accorder à la soirée en se basant sur une **échelle de 1 à 4**, 1 correspondant à un budget faible et 4 correspondant à un budget fort. Grâce à l'API de TripAdvisor, l'application pourra générer un triplet de lieux dont le budget correspondra à au budget **moyen** du groupe. (Détails au point **2.2**)

API utilisés :

- TripAdvisor

2.2. Tests sur des contraintes

2.2.1. La situation géographique

L'application prendra en compte la localisation de chaque personne du groupe. Afin de déterminer une zone qui convient à tout le monde, l'algorithme calcule le centre du cercle circonscrit de diamètre : les deux personnes du groupes les plus éloignées, pondéré par le nombre de personnes présentes dans le demi cercle perpendiculaire à la droite correspondant au diamètre choisi. Ce point sera utilisé pour la recherche du point de rendez-vous au restaurant.

2.2.2. Les horaires de disponibilité

- Afin qu'un utilisateur puisse participer à toutes les étapes de la soirée, il devra renseigner une plage horaire de disponibilité d'au moins 6 heures. En effet, en prenant en compte tous les temps de trajet, plus le temps passé dans chaque lieu sans compter la boîte de nuit, on arrive à un total de 5h. On ajoute une marge d'imprécision d'une heure pour être sûr que le départ d'une personne de la soirée se déroule lorsque le groupe est arrivé en boîte.
- Lors de l'arrivée en boîte, si l'horaire de fin de disponibilité est inférieur à l'horaire de fermeture de la boîte proposée, l'application enverra une notification à l'utilisateur seul, à son horaire de fin de disponibilité, qui redirigera vers un écran contenant son itinéraire retour. En revanche, si l'horaire de fin de disponibilité de l'utilisateur est supérieure à l'horaire de fermeture de la boîte, l'application enverra une notification à l'utilisateur, à l'heure de fermeture de la boîte afin de le renseigner sur son itinéraire de retour.
- Pour éviter que les horaires soient du type 12h-17h, ce qui entre dans les critères des 6h minimum mais ne nous permet pas de proposer de triplet cohérent avec la soirée, nous obligeront à chaque utilisateur de donner une horaire de fin de disponibilité supérieure ou égale à minuit. La plage horaire minimum sera donc 18h-00h.
- Pour éviter que les horaires soient du type 22h-4h, auquel cas l'application ne pourrait pas proposer de triplet cohérent avec les horaires de fermetures des bars et des restaurants, l'horaire de début de disponibilité de chacun devra se situer entre 18h et 20h.

2.2.3. Les goûts de chacun

- Afin de choisir au mieux un restaurant qui plaît à tout le monde, l'application oubliera tout d'abord les restaurant qui correspondent à ce que les différents personnes du

groupe **n'aiment pas**. Ensuite, le choix s'effectuera sur le goût présent dans la **majorité** des usagers dans le groupe de la soirée.

- S'il s'avère qu'il n'y a **aucun** point commun entre les goûts des différentes personnes du groupe, l'algorithme choisira le restaurant ayant la **meilleure note** parmi tous les restaurants correspondant aux goûts de chacun.

2.2.4. Les moyens de transport

Si l'utilisateur vient avec son propre vélo, il **ne pourra pas** suivre le groupe dans les trajets intermédiaires. En revanche, s'il prend un **vélib'**, il n'est pas obligatoire pour l'utilisateur de reprendre le vélo pour effectuer les trajets de groupe. Le trajet en vélo ne sera donc disponible qu'en vélib' afin de faciliter l'implémentation de l'itinéraire.

2.2.5. Budget général du groupe

Les endroits proposés devront correspondre au budget général du groupe qui sera calculé en faisant la **moyenne** des budgets individuels.

2.2.6. La méthode du choix des lieux

Après avoir quitté le bar, les usagers du groupe ne doivent **pas** effectuer une durée de trajet **supérieure à 20 min** pour aller au restaurant, de même pour la boîte de nuit. Ainsi, nous limiterons la recherche du restaurant à **6 km autour du bar** et à **6 km autour du restaurant**.

Lors du choix du lieu, l'application listera tout d'abord les lieux correspondant au goût choisi par la majorité. Ensuite, elle sélectionnera les endroits correspondant au budget général du groupe. L'application devra choisir parmi cette liste l'endroit ayant été le **mieux noté** par les avis Google.

2.3. Fonctions de l'application

2.3.1. La création du profil

Dès le premier lancement de l'application, elle va être confrontée à sa première interaction avec l'interface. Celle-ci lui proposera de créer un profil. Pour que ce profil soit créé, l'utilisateur doit dans un premier temps créer un compte qui permettra de l'authentifier afin de sauvegarder la totalité des paramétrages qu'il aura soumis. Nous allons offrir à notre utilisateur un formulaire de création de compte. Ce formulaire contiendra le strict minimum afin de ne pas lasser l'internaute lors de sa première interaction avec l'application. C'est pour cela que seulement quatre informations seront demandées :

- Un prénom
- Un nom
- Un email
- Un mot de passe

2.3.2. Page d'accueil

Une fois l'utilisateur connecté, il arrivera sur une page d'accueil contenant 4 onglets :

- **Le premier onglet sera celui affiché par défaut.** Il correspondra à l'interface permettant **d'organiser une soirée**. L'interface de cet onglet contiendra une **carte** Google Maps indiquant la **position actuelle** de l'utilisateur lorsqu'il n'est pas en trajet, ou le trajet qu'il est en train d'effectuer avec les informations nécessaires à ce trajet. En dessous de cette carte, il y aura un bouton "Organiser une sortie" qui redirigera vers une page d'invitation de ses amis.
- Le deuxième onglet sera représenté par un signe '+' qui permettra **d'ajouter en ami un utilisateur** contenu dans la base de données des utilisateurs inscrits. En cliquant dessus, on arrive sur une interface dans laquelle l'utilisateur pourra ajouter des personnes à son cercle d'amis ou consulter sa liste d'amis. Dans cette interface il y aura un bouton pour envoyer une demande d'amitié à un utilisateur présent dans la base de données. En cliquant sur ce bouton, une **barre de recherche** apparaît afin que l'utilisateur puisse rechercher une personne de la base des inscrits. L'utilisateur ne pourra pas envoyer de demande d'amitié à un utilisateur avec lequel il est déjà ami. La personne recherchée est affichée sous forme de ligne contenant sa photo, agrandissable en cliquant dessus, et son nom et prénom correspondant à la recherche. En dessous de ce bouton, il y aura la **liste des demandes d'amitiés** reçues par l'utilisateur. L'utilisateur pourra **ajouter l'ami** ayant envoyé la demande ou **refuser la demande**, ce qui aura pour effet la suppression de cette demande de sa liste des demandes. En dessous de cette liste, il y aura la **liste d'amis de l'utilisateur** qu'il pourra dérouler ou réduire. Cette liste permettra à l'utilisateur de supprimer un ami de sa liste en cliquant sur un bouton à côté de chaque ligne contenant la photo de l'ami agrandissable et le nom et prénom de l'ami.
- Le troisième onglet sera représenté par une **bulle de notification**. En cliquant sur cet onglet, on arrive au centre des notifications regroupant tous les types de notifications possible sauf celles des demandes d'amitié. Le comportement de l'application en fonction du type de notification est défini plus bas.
- Le quatrième onglet sera représenté par un **buste et une tête minimalistes**. En cliquant sur cet onglet, on arrive sur une page permettant **d'éditer son profil**. C'est à dire, **modifier les coordonnées de l'utilisateur, ainsi que renseigner ses goûts**.

2.3.3. Page d'invitation d'amis à une soirée

Une fois que l'utilisateur a cliqué sur "Organiser une soirée", il arrive sur une page dans laquelle il peut **inviter à la soirée ses amis** présents dans sa liste d'amis en cliquant sur un bouton '+' présent à côté de chaque ami de la liste. Une fois que l'utilisateur a sélectionné les amis qu'il voulait inviter, il clique soit sur un bouton "Suivant" qui le redirigera vers une page de **préférences propre à l'organisateur**, soit sur un bouton "Annuler" qui le redirigera vers la **page d'accueil**.

2.3.4. Page de préférences générales de la soirée

Une fois que l'organisateur de la soirée a cliqué sur "Suivant" de la "**page d'invitation d'amis à une soirée**", il pourra renseigner, s'il veut, un **thème spécial** pour le bar et choisir la **date** de la soirée. Une fois que l'organisateur de la soirée a choisi les préférences générales de la soirée, il pourra cliquer sur "Valider" qui le redirigera vers la **page de renseignement individuelle** ou sur "Annuler" qui le redirigera vers la **page d'accueil**. Lorsque l'utilisateur clique sur "Valider", l'application envoie une **notification d'invitation** à tous les invités afin qu'ils puissent annoncer qu'ils acceptent ou non l'invitation.

2.3.5. Page de renseignements individuelle

Lorsque que l'**organisateur** de la soirée valide la date de la soirée **ou** qu'un **invité** valide son invitation, il est redirigé sur cette page dans laquelle il pourra renseigner sa **plage horaire** de disponibilités ainsi que son **budget** à accorder à cette soirée. Une fois ces informations renseignées, l'invité est ajouté à la liste des participants de la soirée, qui contient tous les invités ayant accepté l'invitation. Il est par la suite redirigé vers la **page d'accueil**.

2.3.6. Notifications

Il y aura 3 types de notifications :

- **Accepter/refuser une soirée** : dans ce type de notifications, si l'utilisateur clique sur le bouton "accepter" dans la notification, il sera redirigé vers la "**page de renseignements individuelle**". S'il clique sur refuser, il sera redirigé vers la **page d'accueil**.
- **Notification de fin d'heure de disponibilité** : Lorsqu'arrive la **fin de l'heure de disponibilité** de l'utilisateur **ou** qu'il est l'heure de **fermeture de la boîte de nuit**, l'utilisateur reçoit cette notification lui indiquant qu'il est l'heure de rentrer. Lorsque l'utilisateur clique sur cette notification, il sera redirigé vers la **page d'accueil** contenant l'**itinéraire à choisir par l'utilisateur**.
- **Notification de triplet généré** : Lorsque **tous les invités** ont répondu à l'invitation **ou à 16h le jour de la soirée**, les participants de la soirée reçoivent une notification leur indiquant que le triplet a été généré. Lorsque l'utilisateur clique sur cette notification, il est redirigé vers la **page récapitulative du triplet** de la soirée.

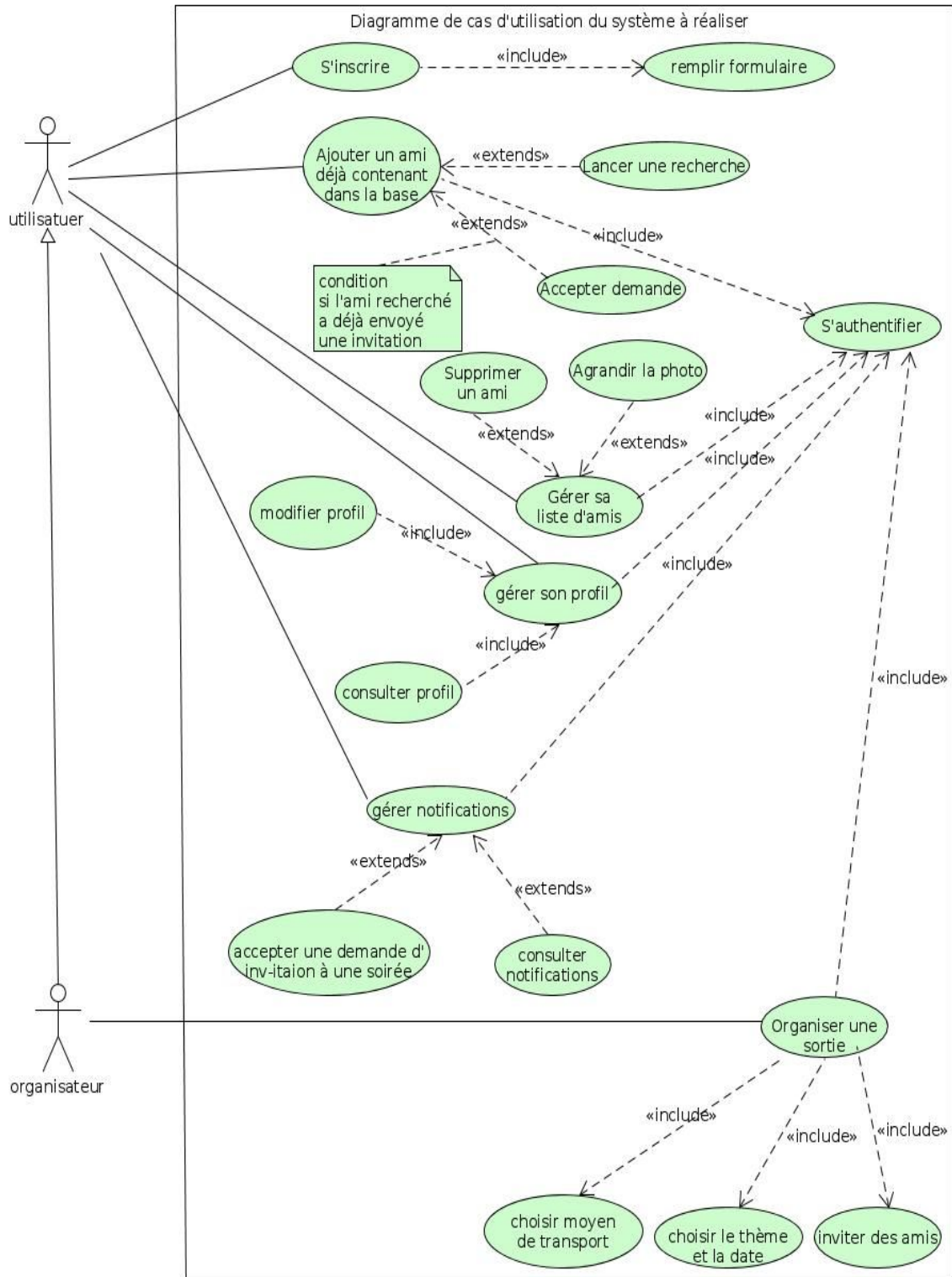
2.3.7. Page récapitulative du triplet

Sur cette page, l'utilisateur pourra voir quels endroits l'algorithme a sélectionné pour la soirée. A côté de chaque endroit, il y aura une courte **description des lieux**, le **budget**, ainsi que la **note** attribuée par les avis Google.

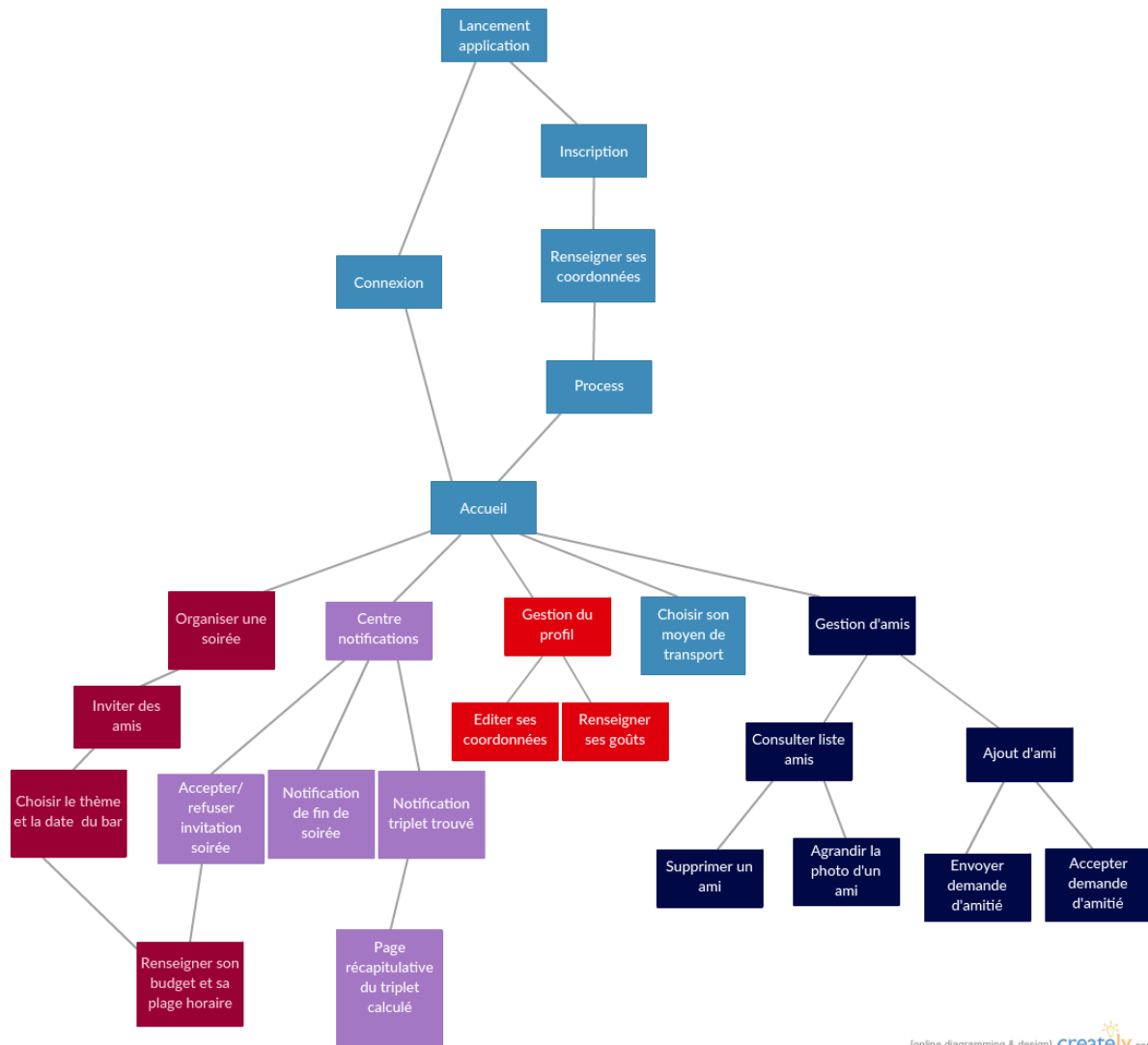
2.3.8. Localisation de l'utilisateur

Chaque smartphone est équipé d'un système de géolocalisation GPS. Nous utiliserons ce système afin de géolocaliser l'utilisateur dans tous ses déplacements, du début de la soirée à la fin de la soirée.

2.4. Diagramme de cas d'utilisation

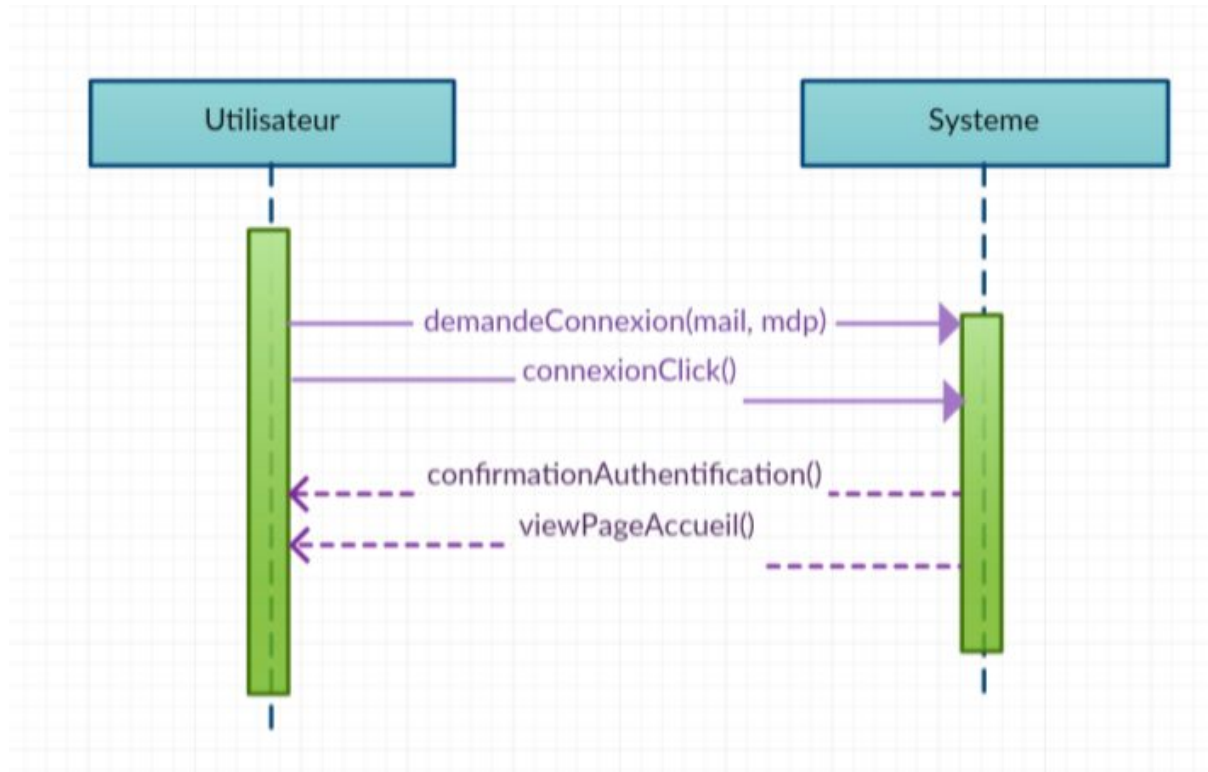


2.5. Arborescence de l'application

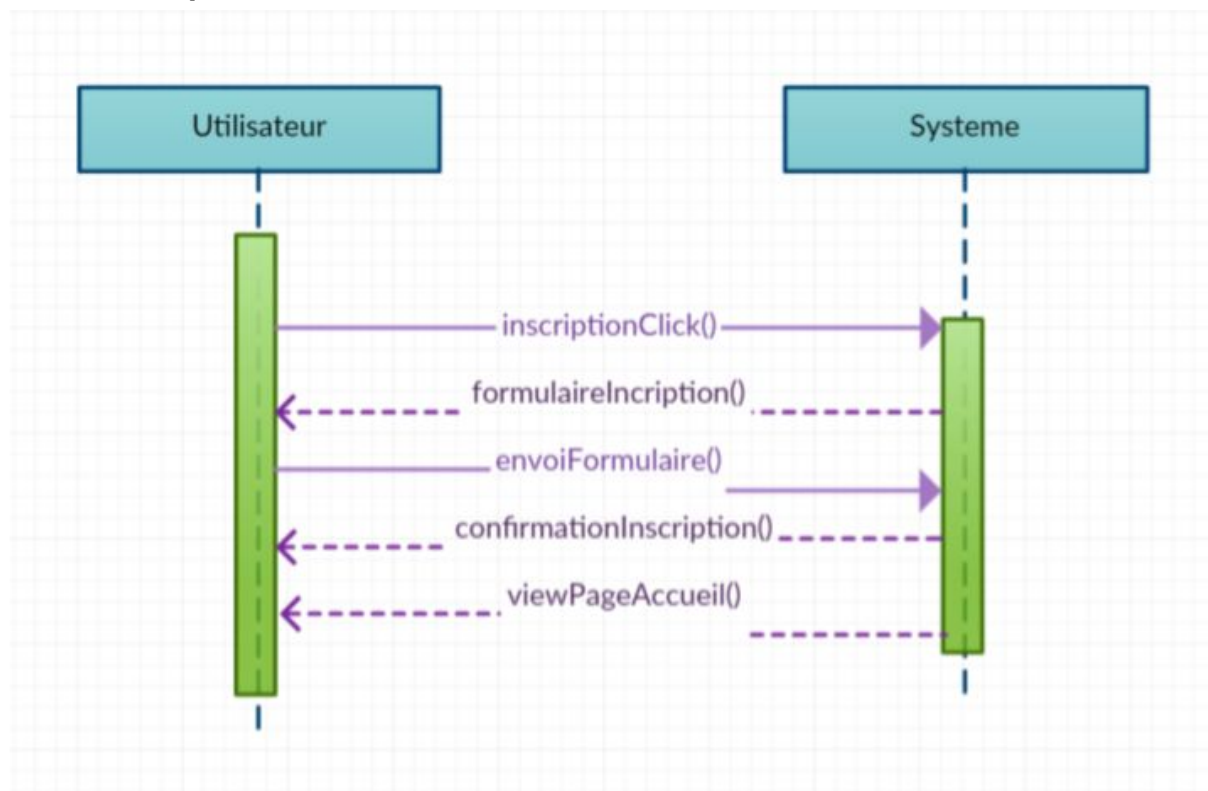


2.6. Diagrammes de séquence

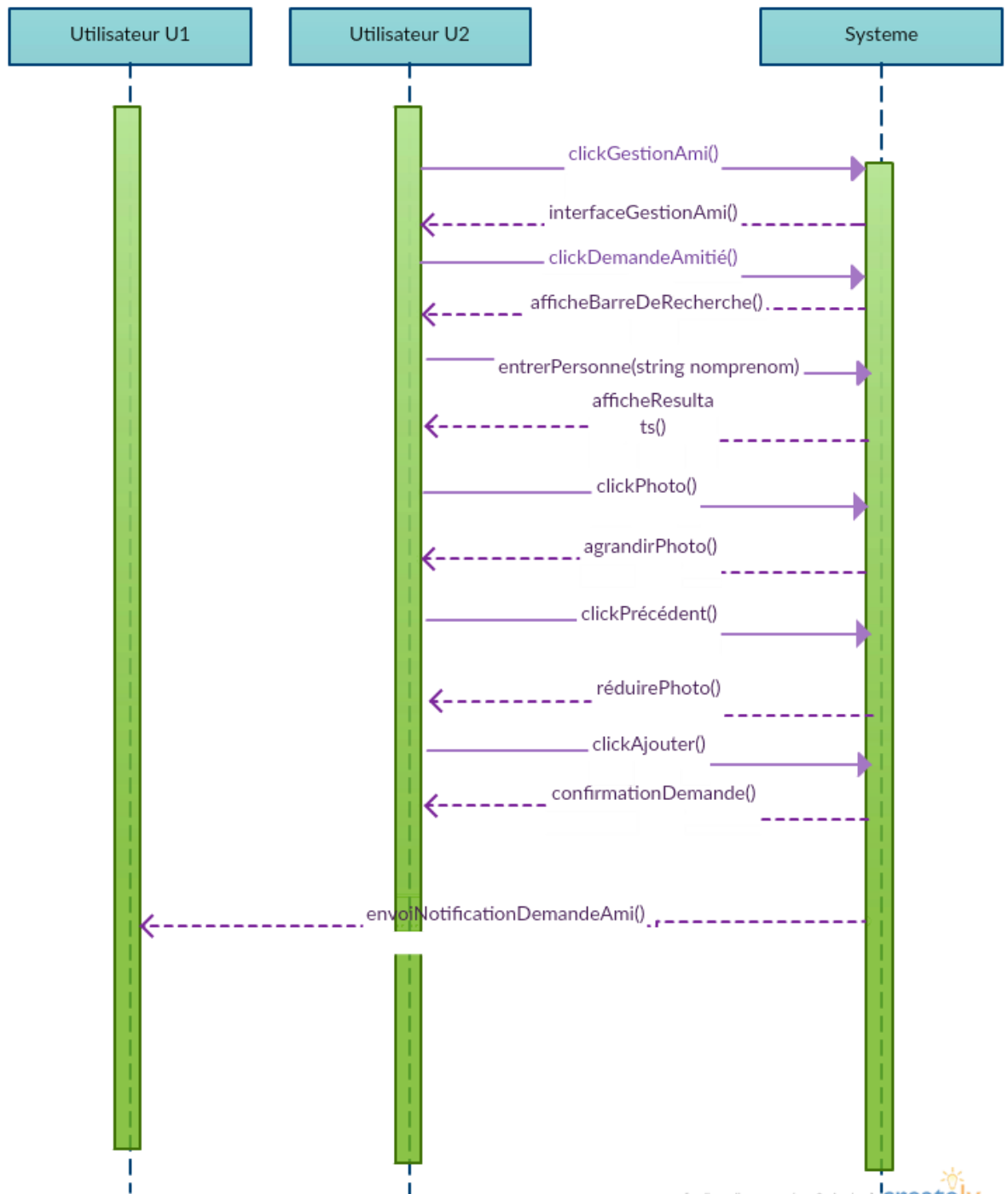
2.6.1. Connexion à l'application



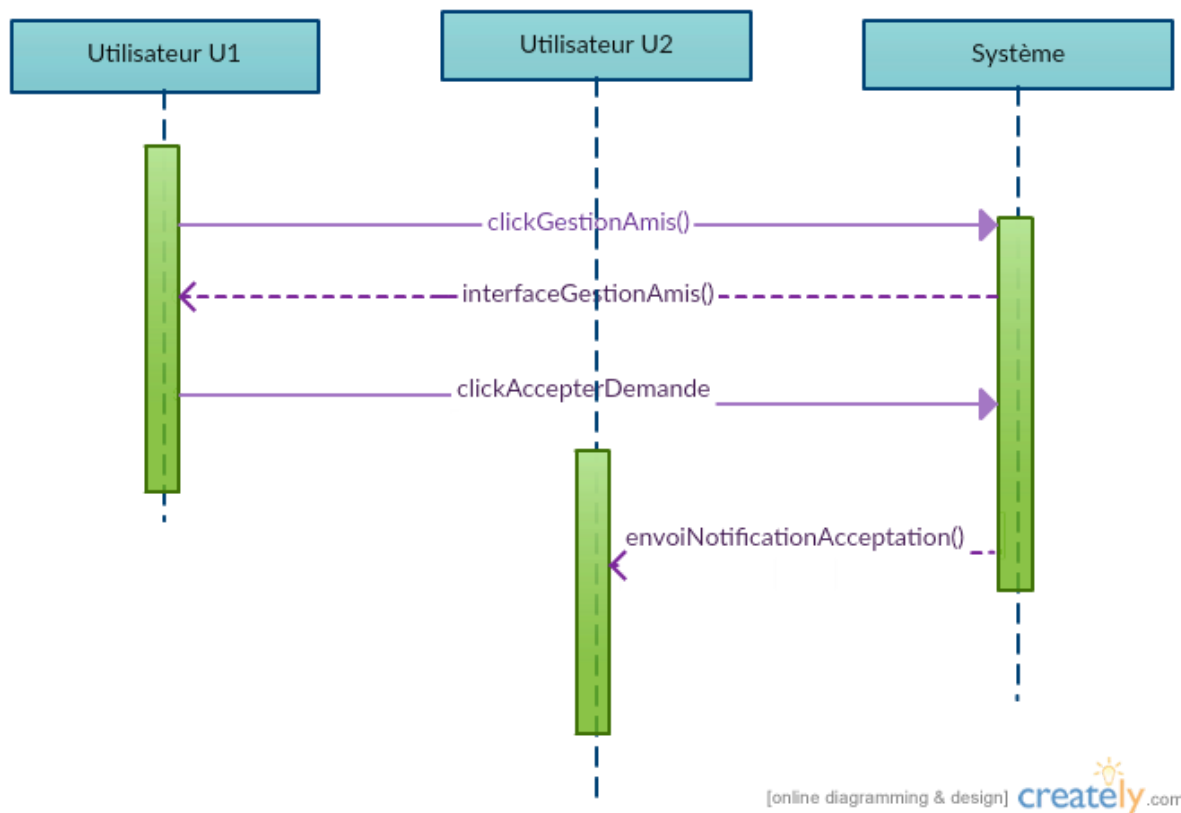
2.6.2. Inscription



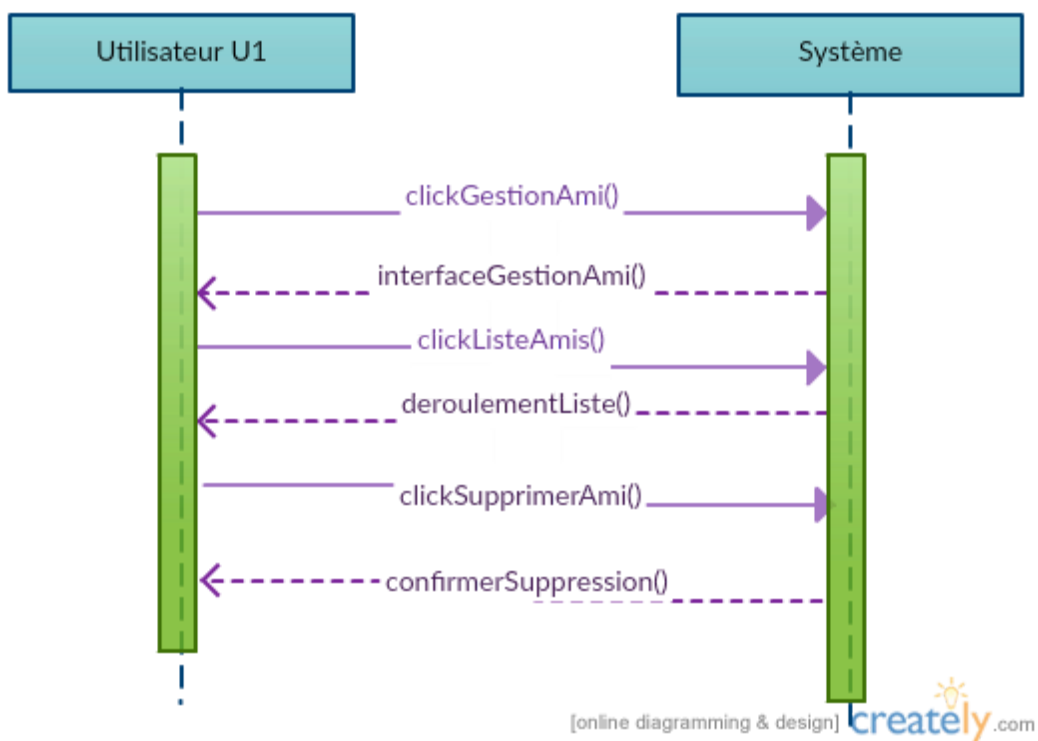
2.6.3. Ajout d'un ami (envoi demande)



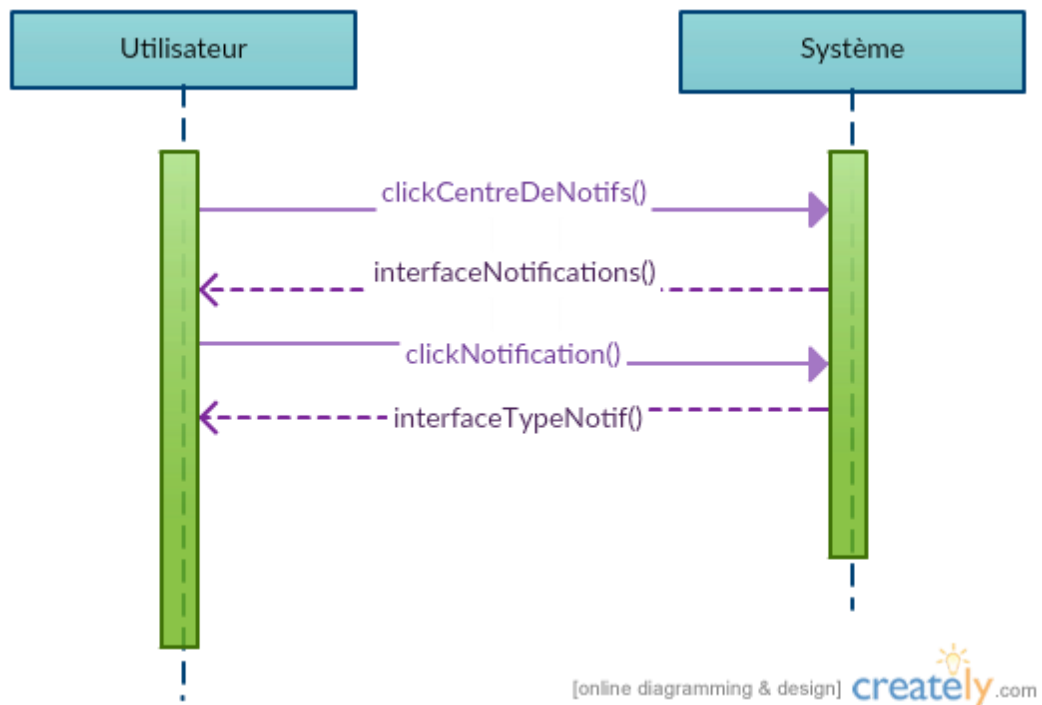
2.6.4. Ajouter un ami (accepter demande)



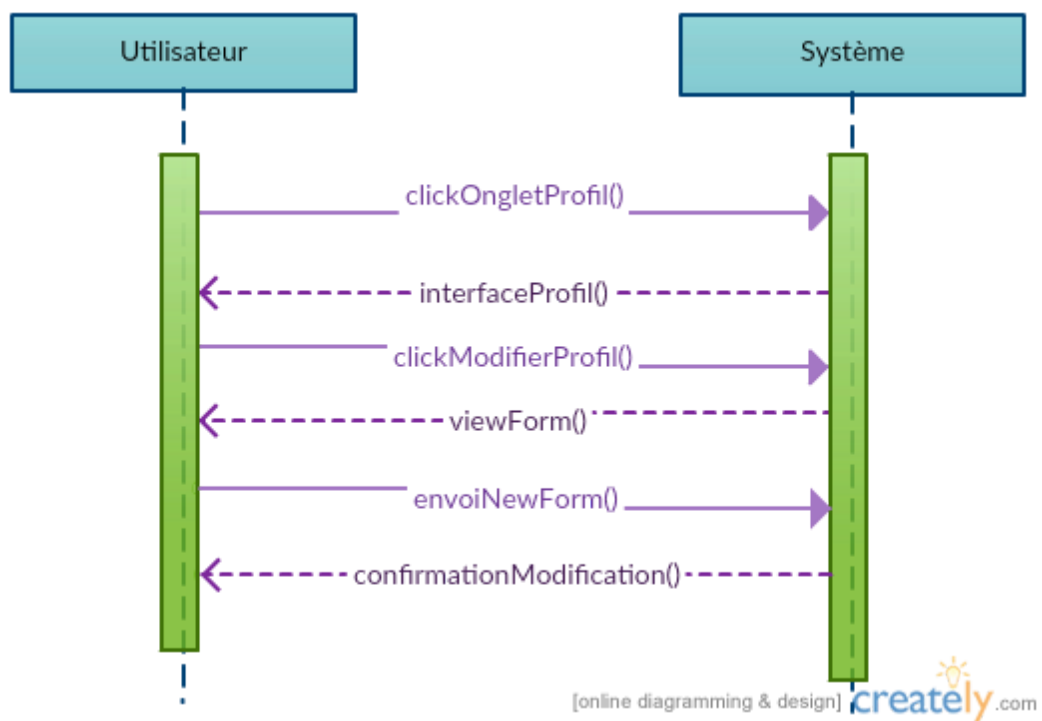
2.6.5. Supprimer un ami



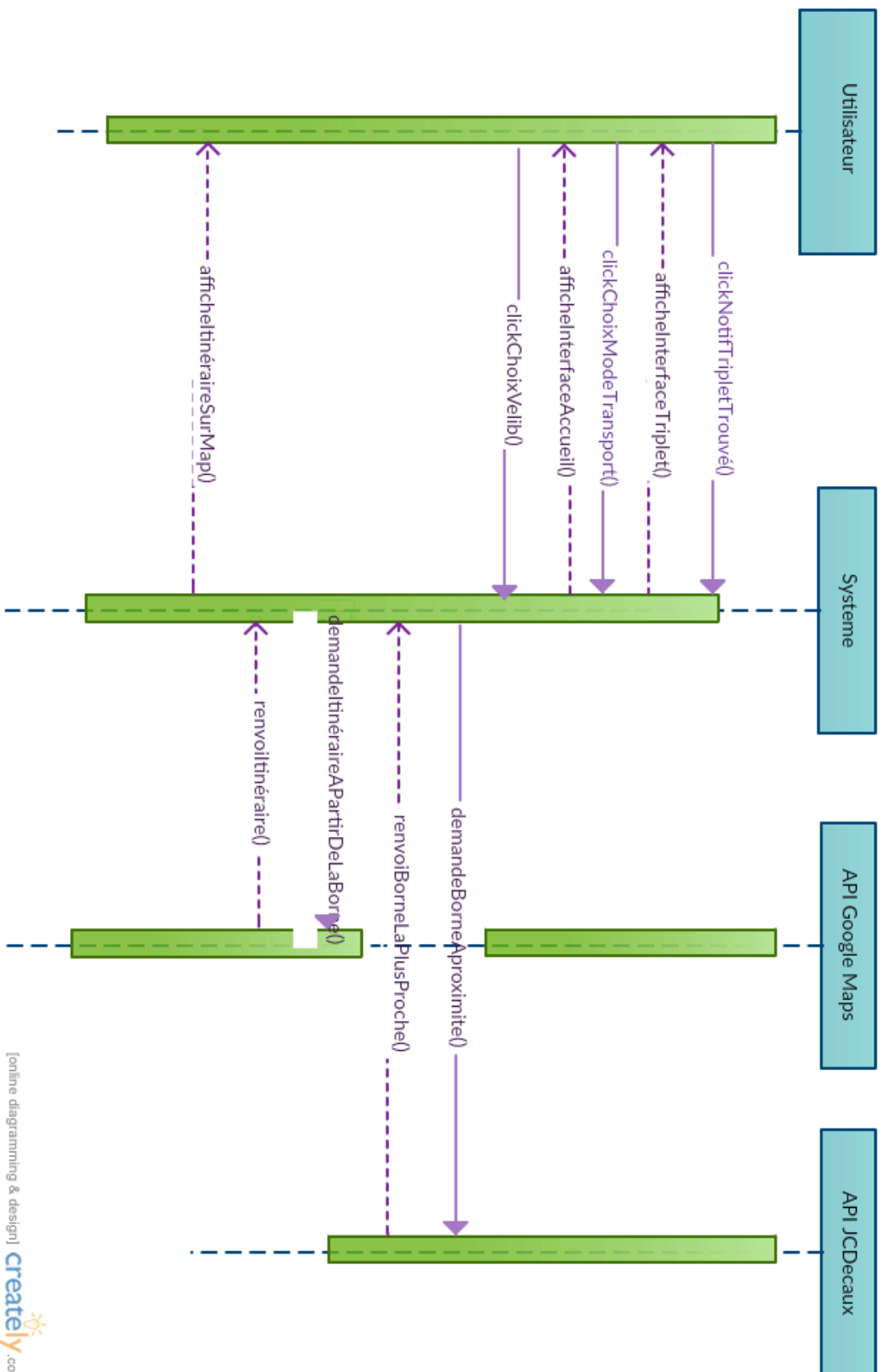
2.6.6. Consulter les notifications



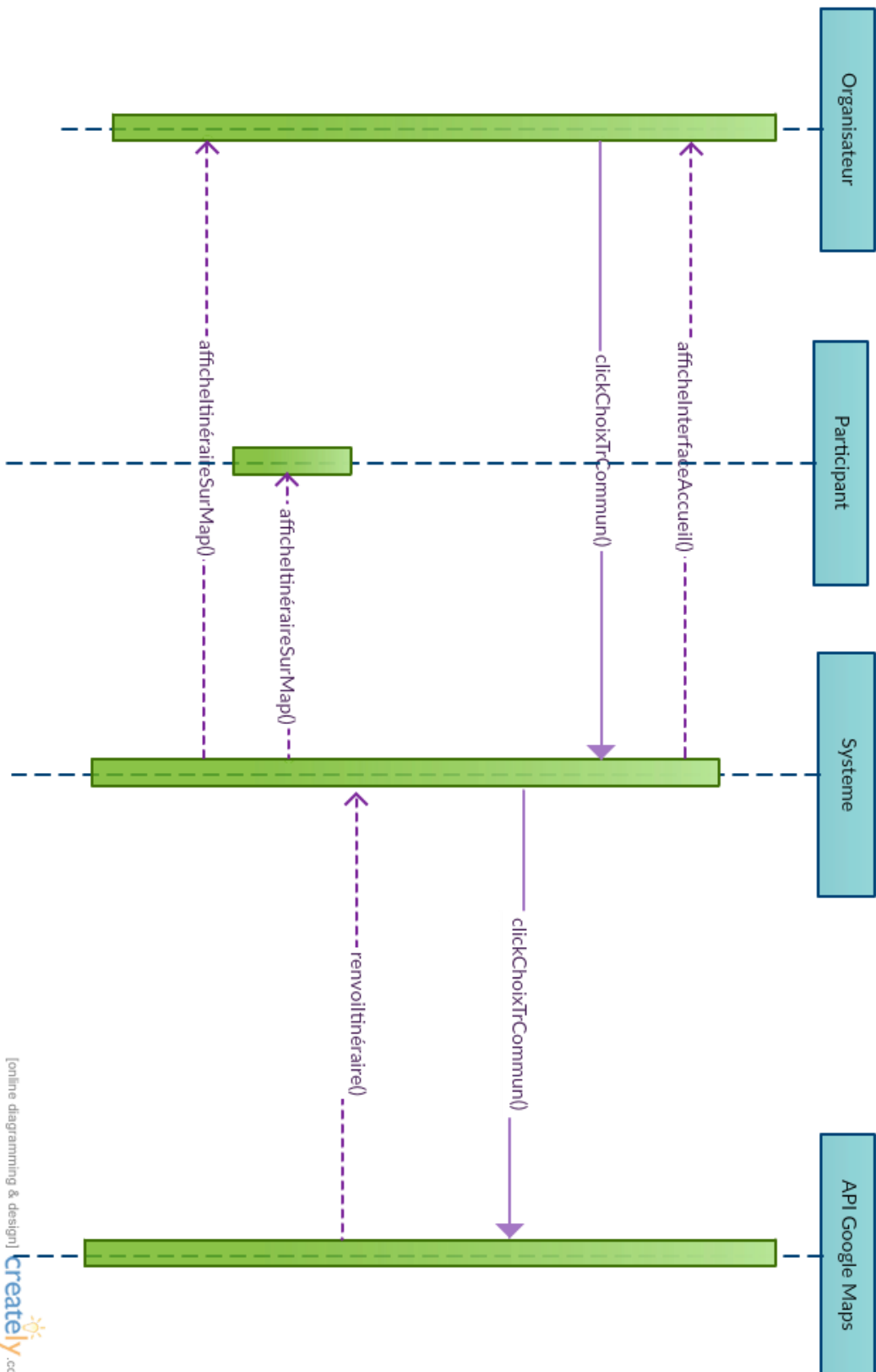
2.6.7. Modifier profil



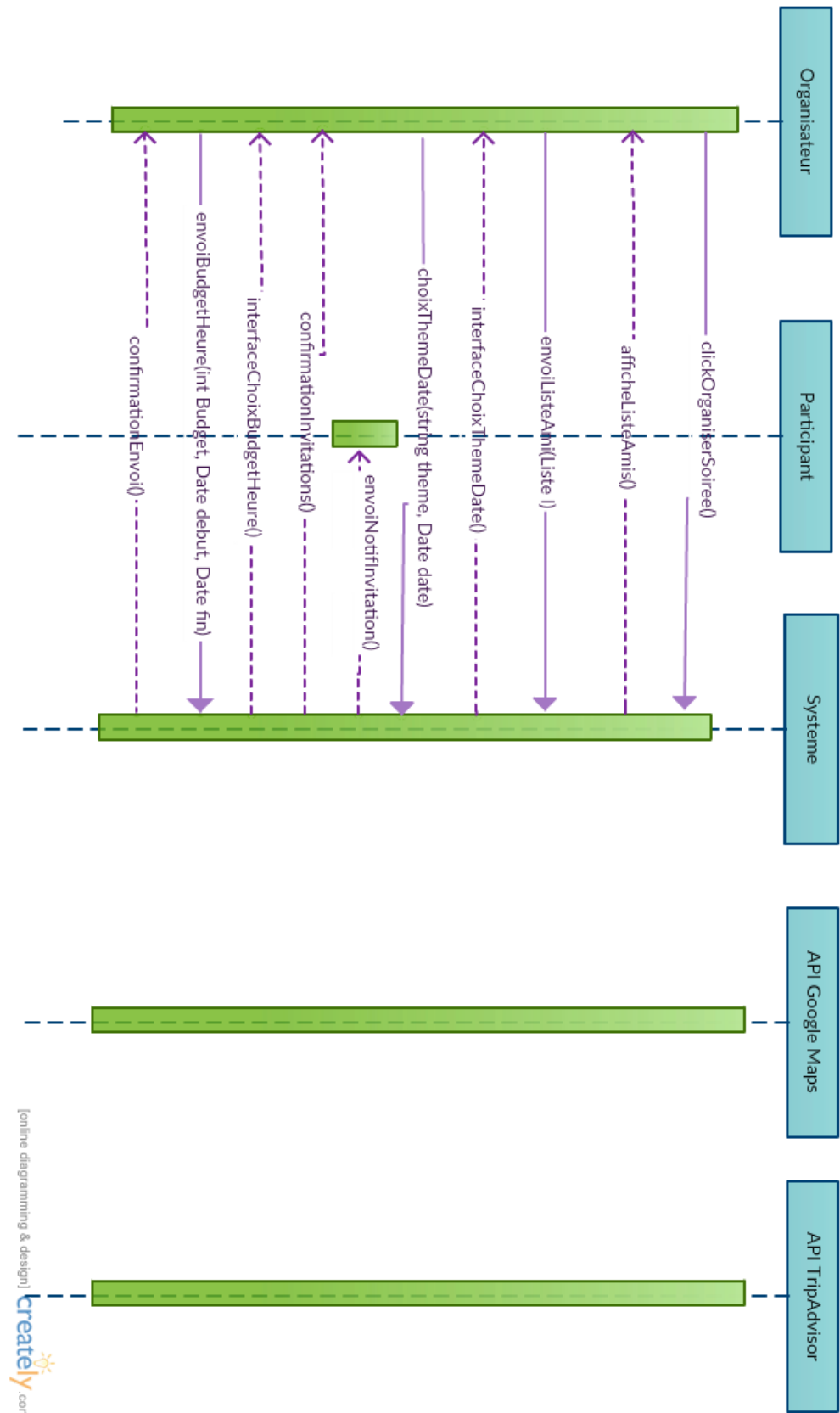
2.6.8. Choix du moyen de transport individuel



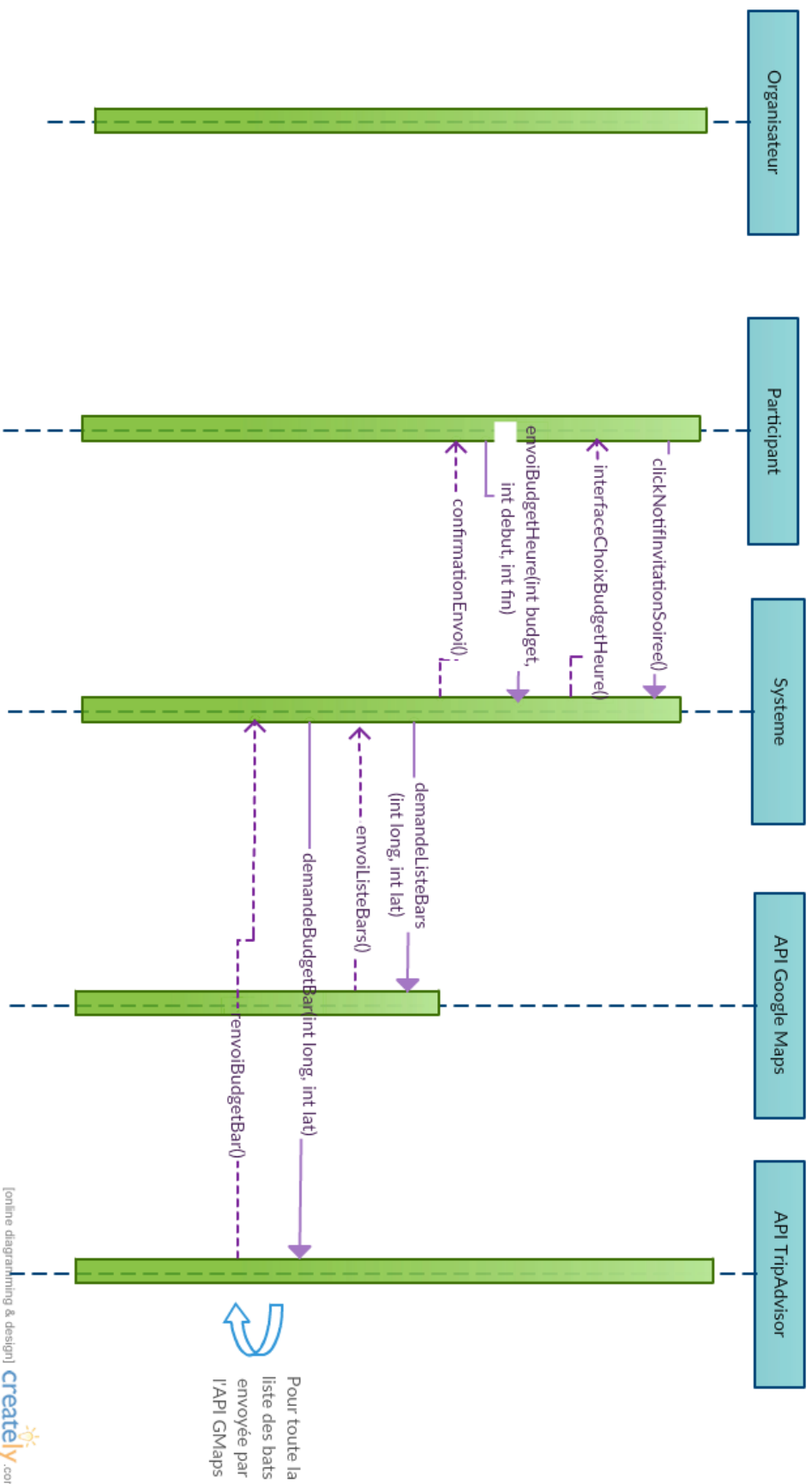
2.6.9. Choix mode de transport groupe



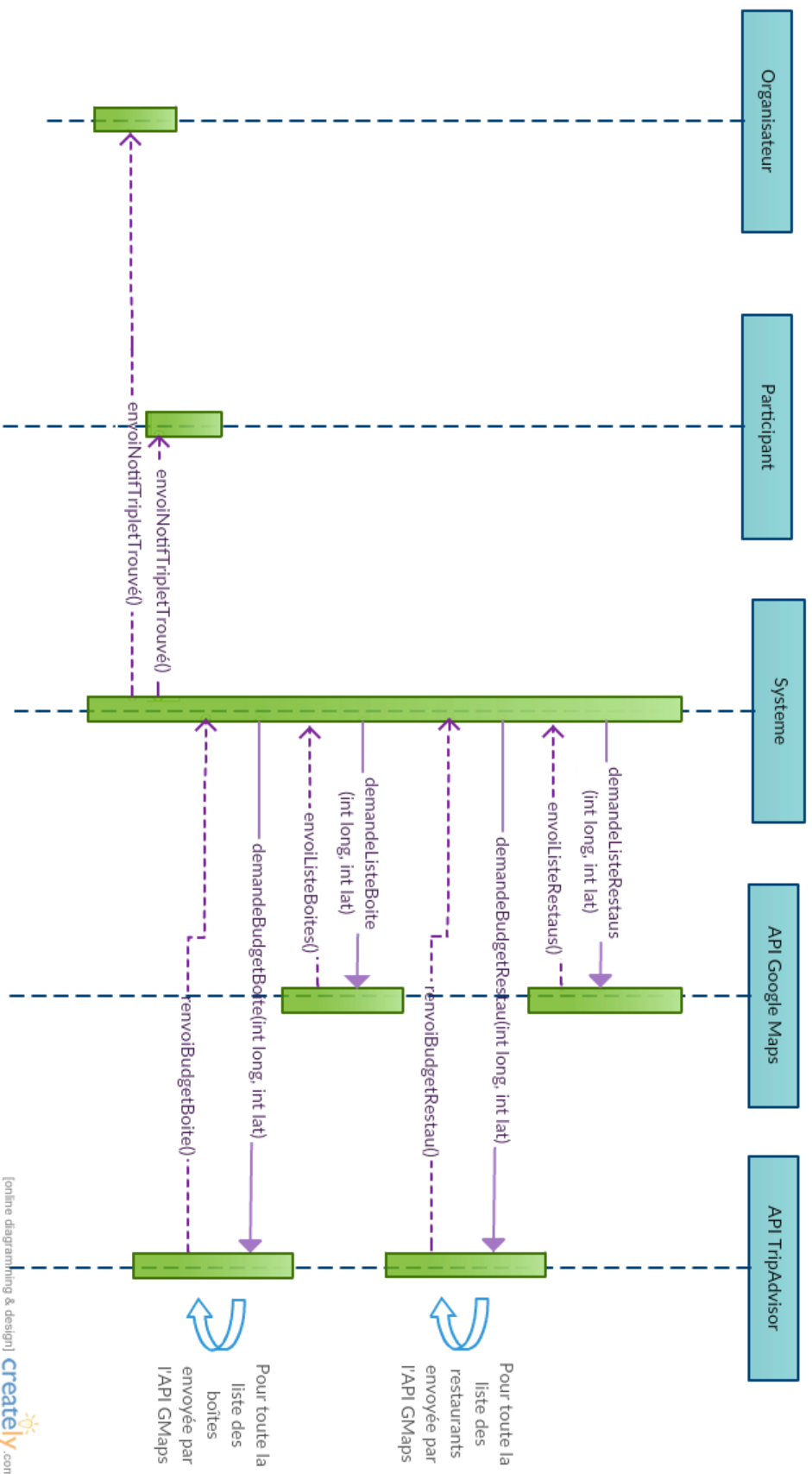
2.6.10. Organiser une soirée



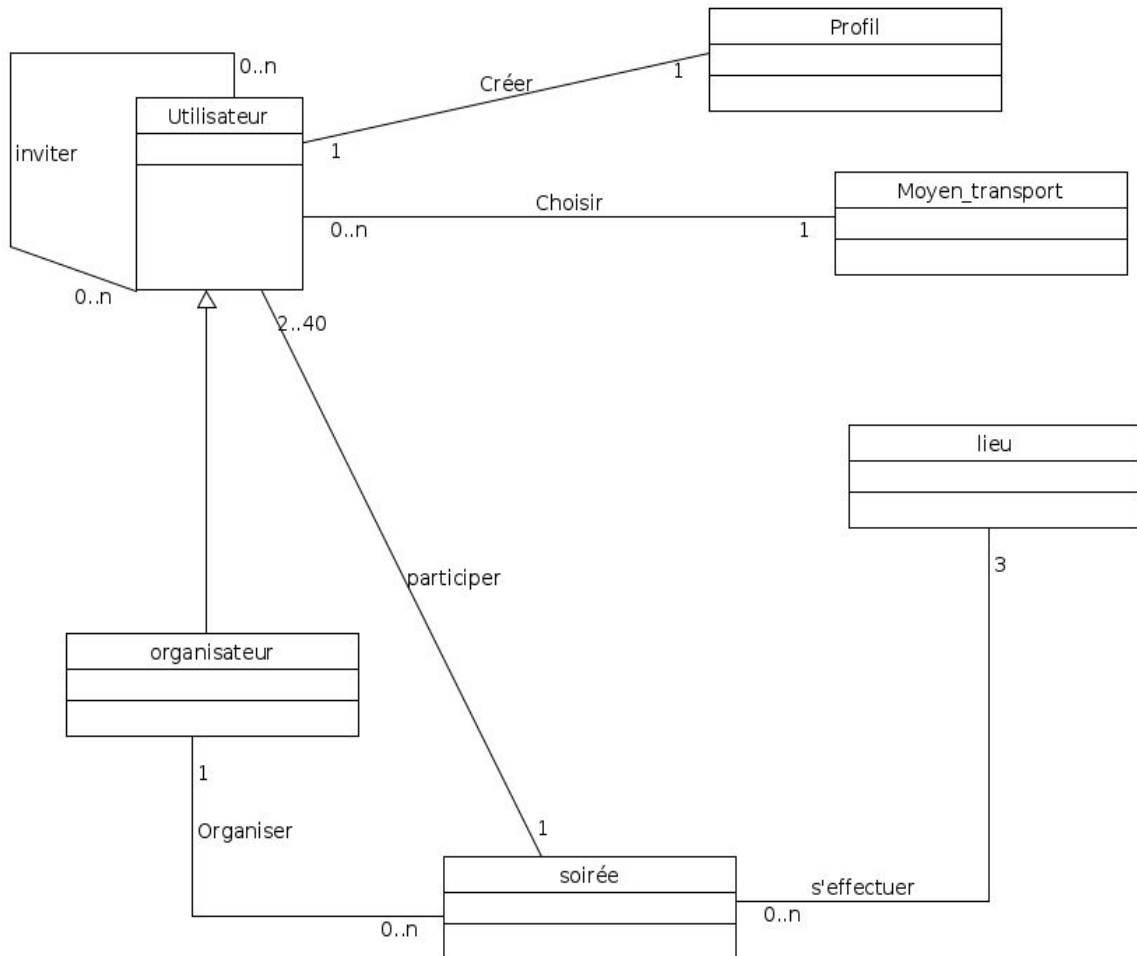
(suite)



(suite)



2.7. Diagramme de classe



3. Préconisations générales

3.2. Charte graphique

3.2.1. Ergonomie

L'utilisation de l'application devra être **intuitive** et **simple**. Les différentes interfaces devront être **épurées** et agréables à l'oeil. Elle ne devront **pas** comporter **plus de trois boutons** de redirection.

3.2.2. Gamme de couleurs

Les couleurs de l'application devront être **flashies mais agréables à l'oeil**. Les boutons de validation et de suite seront en **vert**. Les boutons d'annulation seront en **gris clair**. L'arrière plan des interfaces sera **blanc ou vert**.

3.2.3. Police

La même police sera utilisée dans toute l'application. Par souci de neutralité et parce qu'il faut qu'elle soit agréable à l'oeil, l'ensemble des zones de texte seront en **Verdana**.

3.3. Développement

Nous avons choisi de développer l'application entièrement sur **Android**. En effet, les développement d'une application mobile est plutôt facile d'accès en raison d'un nombre considérable de ressources présentes sur internet. Les API mis à disposition permettent aussi de faciliter la tâche de développement. En effet, elles sont très complètes et faciles d'utilisation.

3.4. Matériel et compétences

L'application sera entièrement codée en **Java pour Android**. Nous utiliserons les API suivants :

- Google Maps API
- JCDecaux API
- TripAdvisor API

Les fichiers en format **JSON** seront traités par la bibliothèque JSONObject de Java. Nous utiliserons l'**IDE Eclipse** afin de faciliter la programmation.

L'interface graphique sera implémentée grâce au logiciel **Android Studio Bundle**.

4. Modalités

4.1. Analyse des besoins

Il est nécessaire de bien comprendre le vocabulaire employé afin de pouvoir l'expliquer à ceux qui vont participer à cette analyse. Analyser un besoin c'est devenir un « traducteur », cette traduction doit être la plus fidèle possible afin que celui qui va « couvrir » le besoin y réponde précisément. Le besoin est ensuite décrit dans un document final qui peut être appelé « cahier des charges client » ou « cahier des charges utilisateur ».

4.2. Phases du projet

Le projet est découpé en trois principales phases :

- **Cahier des charges et analyse des besoins**, première phase de réalisation d'un projet, conditionne sa réussite dans la mesure où elle définit les besoins réels de ceux qui vont utiliser le résultat final. Phase de communication et d'échange, elle est souvent le reflet du résultat final. Nécessitant rigueur et méthode, c'est l'une des phases les plus difficiles de la conduite de projet, cette phase consiste à définir le système à réaliser et elle doit être rendu avant 19/02/2016.
- Deuxième phase qui est **le cahier de conception** qui consiste à diviser le produit en sous-ensembles tout en s'assurant que la solution réalise toutes les fonctions prévues, permet de contrôler le bon fonctionnement et qui valide le produit et qui sera rendu avant le 26/03/2017.
- Troisième phase qui est **l'implémentation** où chaque fonctionnalité définie dans la phase de capture des besoins c'est à dire la première phase devra être réalisé dans la phase d'implémentation et cela avant le 03/05/2017.

4.3. Réception des résultats de prestation

Tout le travail effectué par le **prestataire** (livrable) sera soumis au **client**, qui vérifiera sa **conformité** au cahier des charges.

Modalités de validation des documents : Sont concernés par cette section tous les documents fournis par le prestataire au cours de la prestation : **étude de faisabilité, spécifications détaillées, cahier de recette, et guide d'utilisation.**

Après **remise** de ces éléments pour validation, le client disposera d'un **délai de 2 semaines** pour les valider.

A l'issue de ce délai le client peut soit **valider** ou **refuser**. Si le client valide le document, le prestataire sera **autoriser** à passer à l'étape **suivante de la prestation**. Le refus correspond à une **non-conformité** ou **incomplétude** du livrable par rapport à sa définition ou aux besoins du client.

Pour les livrables refusés, le prestataire disposera d'un **délai de 2 semaines** pour répondre à celles-ci et présenter les **documents modifiés**.

Ce même délai sera utilisé par le client pour donner une **nouvelle validation**.

Une fois le développement de l'application **terminé**, il faudra procéder à la validation du **bon fonctionnement** de celle-ci. Cette phase est importante et demande une forte implication de la part des intervenants chargés de réaliser cette mission. En effet, celle-ci permettra de vérifier que les modules livrés fonctionnent correctement et qu'ils **répondent aux besoins** défini par le cahier des charges. Une fois la phase de conception validée, la mise en production de l'application débutera. Si aucune remarque écrite n'est formulée, nous pourrons considérer que les modules livrés sont **conformes**. Si des dysfonctionnements apparaissent dans **les six mois** qui suivent la mise en ligne de l'application, le prestataire se devra de corriger **les erreurs décelées**.

5. Sources

Construction du cahier des charges :

- <http://gaetano-notorio.com/portfolio/cahier-des-charges-pour-une-application-mobile/>
- Divers cahier des charges fournis par les proches : exemple d'un projet d'application CGI et exemple de cahier des charges de Publicis Technology

Documentation des API :

- <https://console.developers.google.com/?hl=FR>
- <https://developer-tripadvisor.com/content-api/>
- <https://developer.jcdecaux.com/>