



HAMAOUI Djenna
GUERID Yousra
RAYNAL Swann



Projet Génie Logiciel Avancé L3

[*Cahier de conception de DYSAndFriends*]



Professeurs :
NGUYEN VAN Hai
WOLFF Burkhardt

Année 2017-2018

L3 Informatique /MIAGE

Table des matières

1 Introduction

- 1.1. Caractéristiques du système visé
- 1.2. Dépendances

2 Structure globale

- Base de données

3 Démarrage

4 Création d'une sortie

- Initialisation de la sortie
- Ajout de participants
- Choix des étapes
- Choix des contraintes
- Calcul de la sortie idéale

5 Carte

6 Tests

- Tests unitaires
- Tests fonctionnels
- Tests d'acceptation

Table des figures

Figure1_ Diagramme des classes principales

Figure2_ Diagramme entités-relations

Figure3_ Diagramme de séquence de l'accueil

Figure4_ Diagramme de séquence d'initialisation de sortie

Figure 5_ Diagramme de séquence d'ajout d'un participant

Figure6_ Diagramme de séquence de modification des étapes

Figure7_ Diagramme de classe du graphe

Figure8_ Diagramme de séquence de calcul de la sortie

1. Introduction

Cette application a pour but d'offrir à ses utilisateurs un moyen automatisé et simple pour organiser une soirée. Après avoir donné les participants, la date, les contraintes et la suite d'étapes que la soirée devra comprendre, l'application web se chargera de trouver un parcours passant par toutes ces étapes en respectant les contraintes (qu'il s'agisse des horaires ou de contraintes données par l'utilisateur).

1.1. Caractéristiques du système visé

On utilisera Java comme langage de programmation.

Les ressources sont très limitées. Il faudra donc consommer le moins de mémoire RAM et physique possible. L'application devra aussi avoir une interface fluide et agréable avec un contenu adapté aux différents formats d'écrans.

1.2. Dépendances

Les APIs Google Places et Google Maps seront nécessaires car exploitées pour trouver les différentes étapes de la sortie, pour afficher une carte et transformer les adresses en coordonnées GPS (latitude, longitude...).

Aussi, l'utilisation d'une base de données interne sera primordiale pour sauvegarder et manipuler nos données. On utilisera pour la base le langage SQL et le SGBDR MySQL.

2. Structure globale

Les classes dont sera constituée l'application sont représentées par le diagramme de la figure 1.

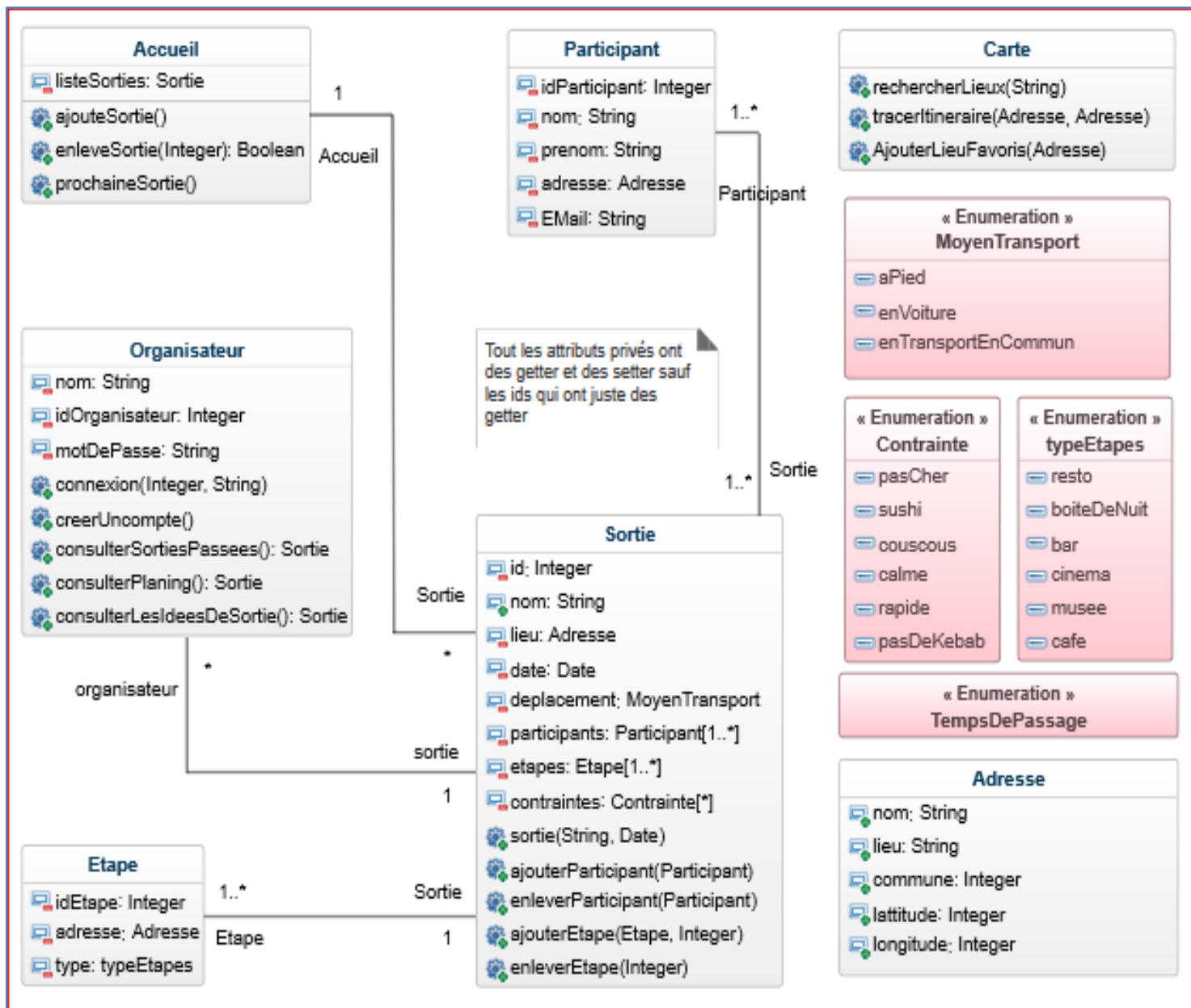


Figure 1 – Diagramme présentant les classes (et leurs relations) telles qu'elles seront implémentées dans notre application

a. Base de données

La base de données sera nécessaire pour stocker en local toutes les informations qu'il serait utile de garder. Cela peut concerner les idées de sortie, les différentes soirées déjà organisées, mais aussi pour exploiter plus proprement d'autres données telle que la liste des étapes possibles où leur durée.

Sa structure sera assez proche de celle des classes afin de rendre sa manipulation la plus logique possible.

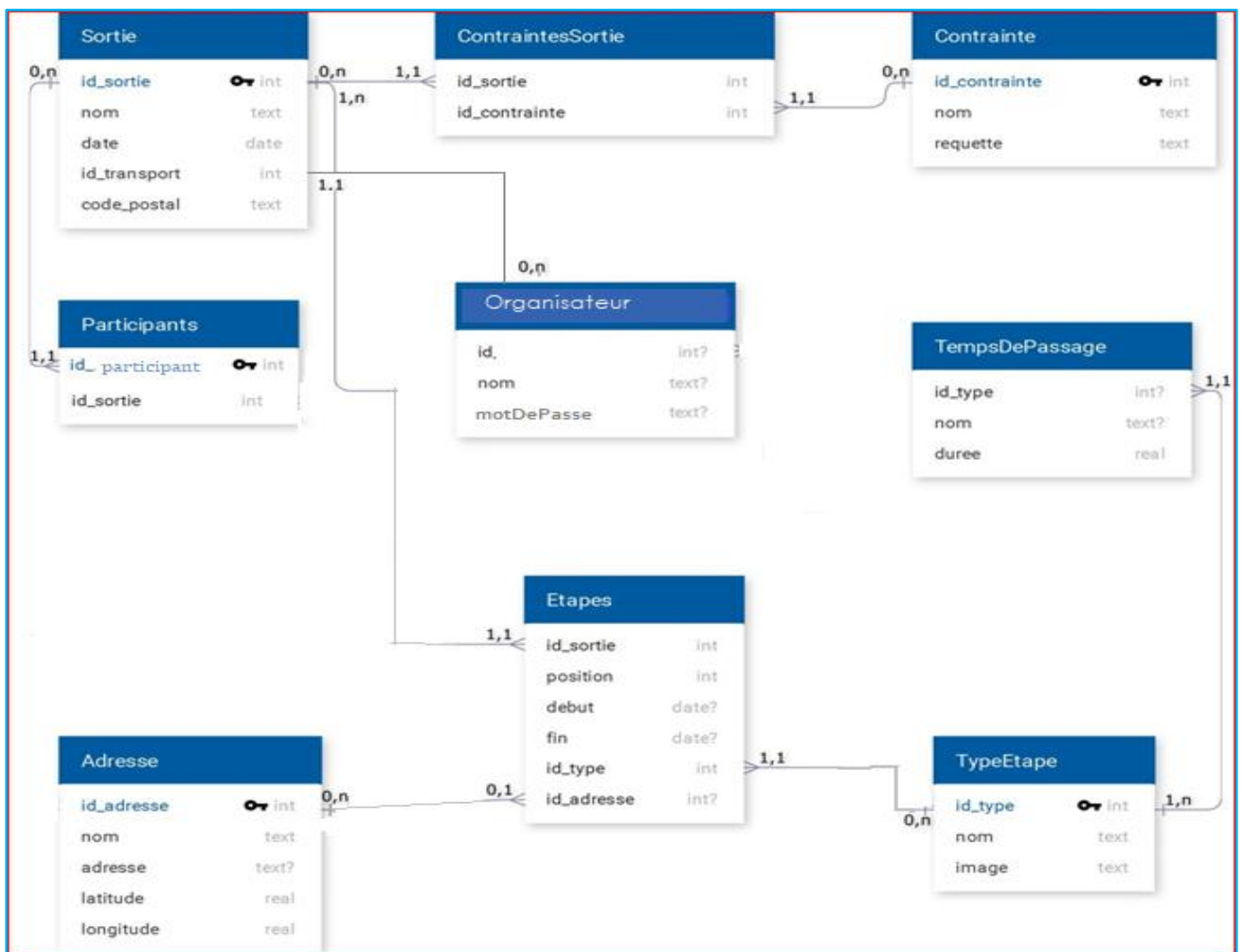
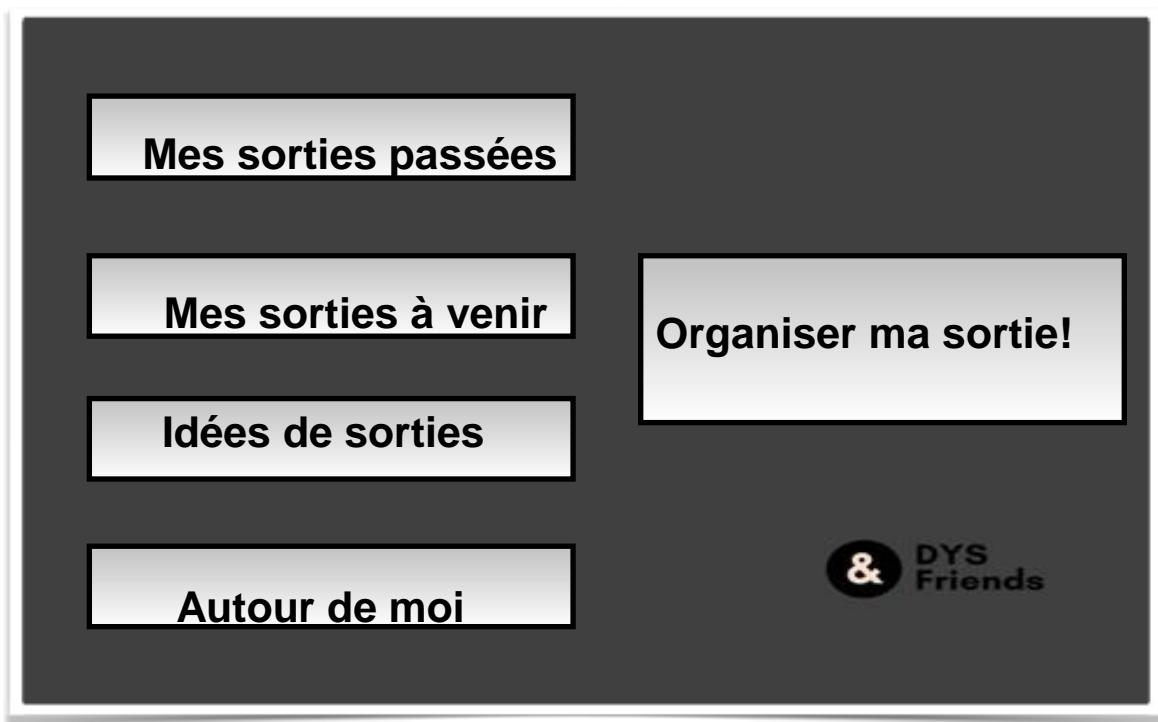


Figure 2 – Diagramme entités-relations présentant la structure de la base de données

3. Démarrage

Le démarrage après la connexion de l'utilisateur de l'application est présenté dans le diagramme3.

La page d'accueil qui donnera un aperçu des soirées à venir (ou en cours) ainsi qu'un accès rapide à toutes les fonctions de l'application.



Accueil de l'application web

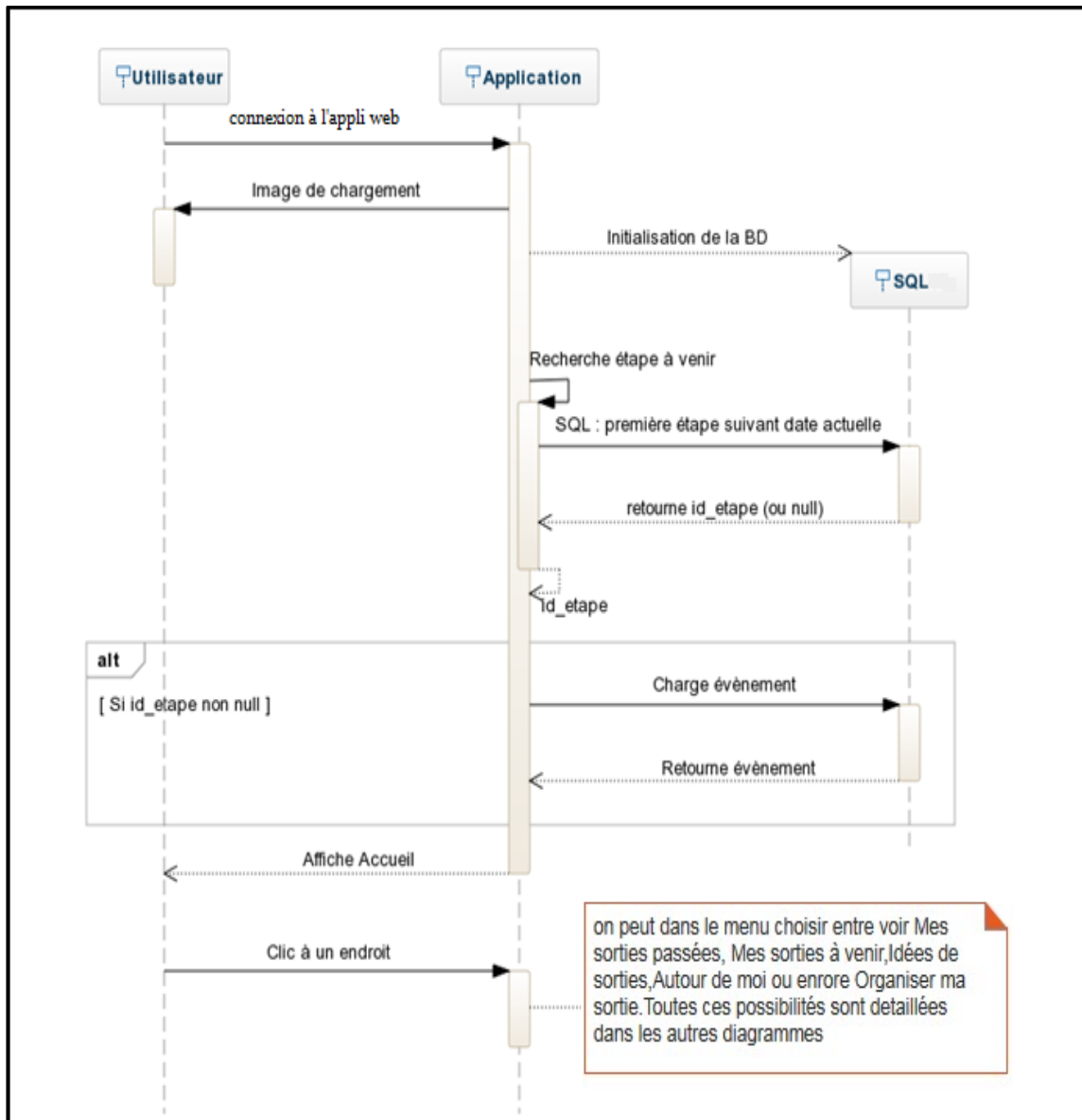


Figure 3 – Diagramme de séquence représentant le déroulement interne de l'application lors de son démarrage

4. Création d'une sortie

La création d'une sortie s'effectue en plusieurs étapes qui seront détaillées séparément dans les sections suivantes.

a. Initialisation de la sortie

Il est tout d'abord nécessaire d'initialiser la sortie ; c'est à dire donner les informations de base qui permettront de la différencier des autres : un nom, un lieu, une date, un moyen de transport et des participants (ce dernier point sera traité dans la prochaine section).

L'interface est donc très simple tout comme son fonctionnement (figure4).

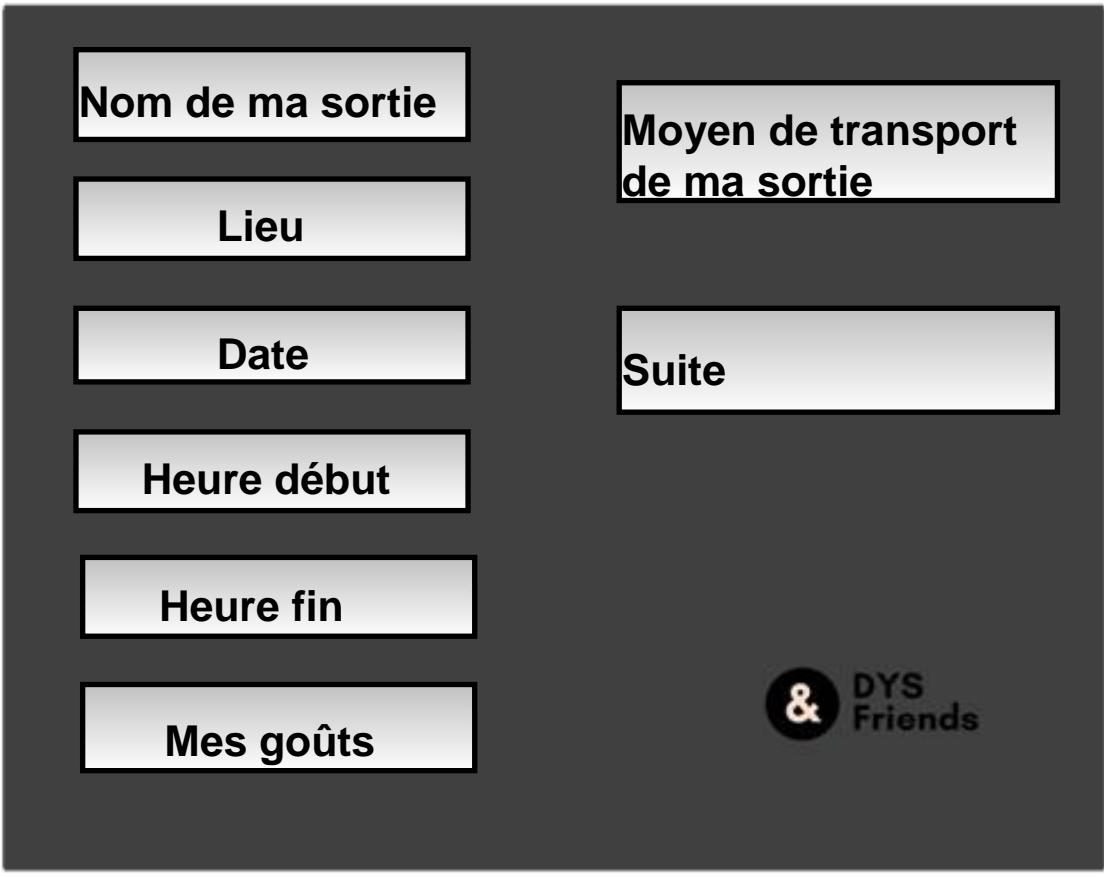


Figure 4 : Interface de création d'une sortie. L'interface est présentée sur un fond gris foncé. Elle comporte deux colonnes de champs de saisie. La colonne de gauche contient six champs empilés : 'Nom de ma sortie', 'Lieu', 'Date', 'Heure début', 'Heure fin', et 'Mes goûts'. La colonne de droite contient deux champs : 'Moyen de transport de ma sortie' et 'Suite'. En bas à droite, il y a un logo avec un ampersand dans un cercle noir, suivi du texte 'DYS Friends'.

Interface permettant d'initialiser une nouvelle sortie

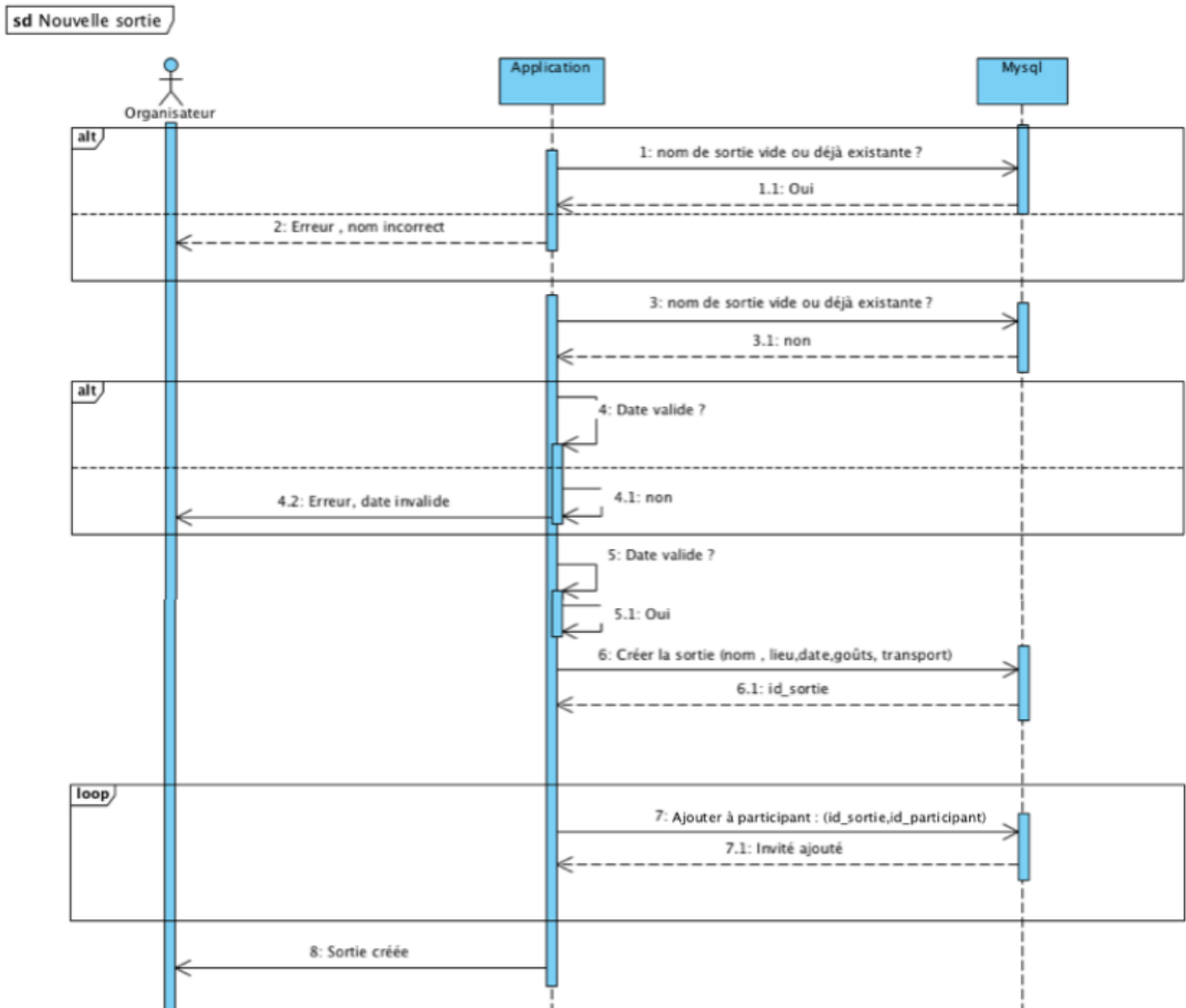


Figure 4 – Diagramme de séquence représentant les étapes et les cas d'erreur lors de l'initialisation d'une nouvelle sortie

b. Ajout de participants

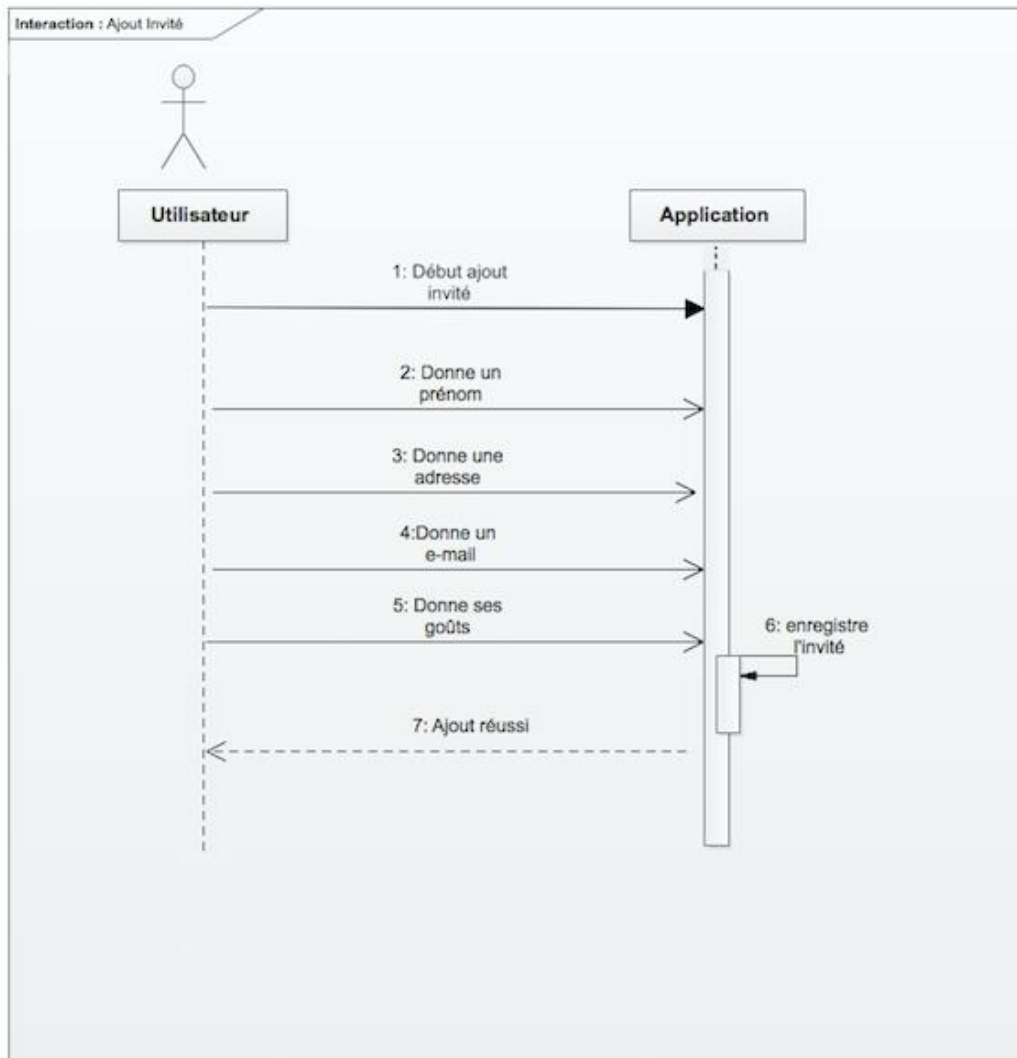


Figure 5 –Diagramme de séquence montrant comment est ajouté un participant en renseignant la totalité des informations possibles sur cette personne.

c. Choix des étapes

Vient ensuite l'étape permettant de définir les étapes de la sortie ainsi que leur ordre. Pour cela, l'interface ci-dessous affichera les différents endroits possibles qu'il suffira de cliquer sur le plus pour choisir dans l'ordre voulu.

Tout cela est détaillé dans le diagramme6.

The screenshot shows a mobile application interface with a dark grey background. At the top, the title 'Où aller' is displayed in white. Below it, there is a text input field containing 'Bar,Restaurant,Boite de nuit' and a blue circular button with a white plus sign. Underneath, the section 'Liste des' is followed by a scrollable list of three items: 'Aller au bar', 'Manger au restaurant', and 'Aller en boite de nuit'. At the bottom center is a large white button labeled 'Valider'. In the bottom right corner, there is a logo for 'DYS Friends' featuring an ampersand inside a circle.

Page d'ajout de lieux

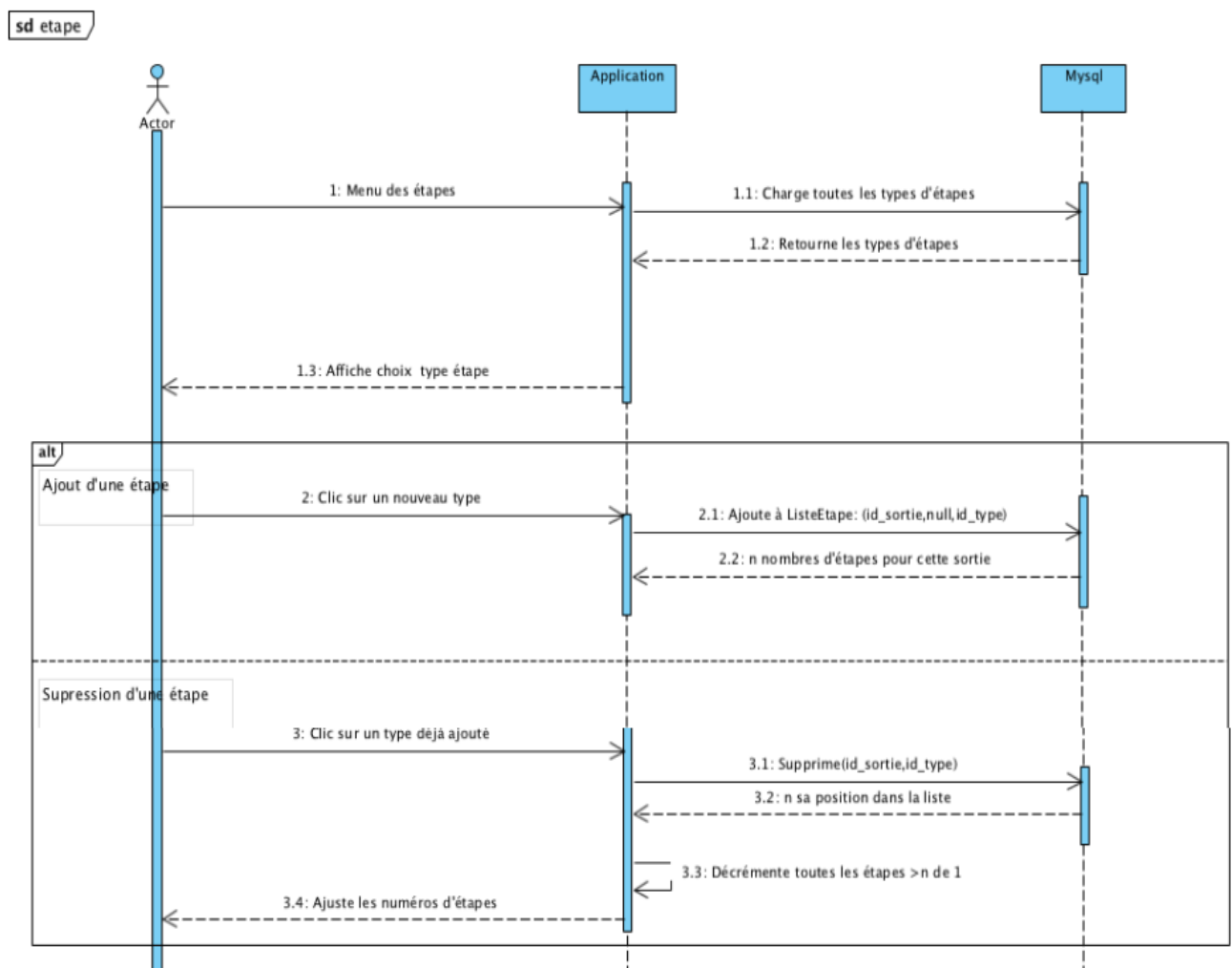


Figure 6 – Diagramme de séquence représentant comment ajouter ou enlever une étape à la sortie

d. Choix des contraintes

Les contraintes seront choisies parmi une liste de possibilités prédéfinies. Il sera possible d'en sélectionner autant que l'on veut.

e. Calcul de la sortie idéale

Le calcul de la sortie passant par la récupération d'adresses grâce à l'API de Google, une connexion internet sera obligatoire dans cette partie du programme.

L'algorithme utilisé est représenté par le diagramme de la figure 8 et consiste simplement en un algorithme de parcours en largeur d'un arbre. Cet arbre sera représenté à l'aide des classes de la figure 7, et sera construit ligne par ligne, en prenant bien en compte l'heure qu'il sera une fois arrivé à cette étape, l'heure qu'il sera une fois cette étape terminée, ainsi que toutes les contraintes que l'utilisateur aura éventuellement spécifié.

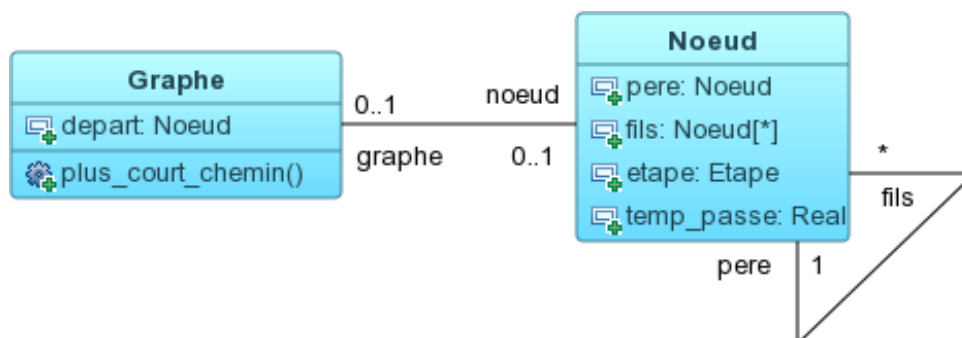


Figure 7 – Diagramme de classe utilisé pour représenter un graphe orienté d'étapes

sd SortieFinal

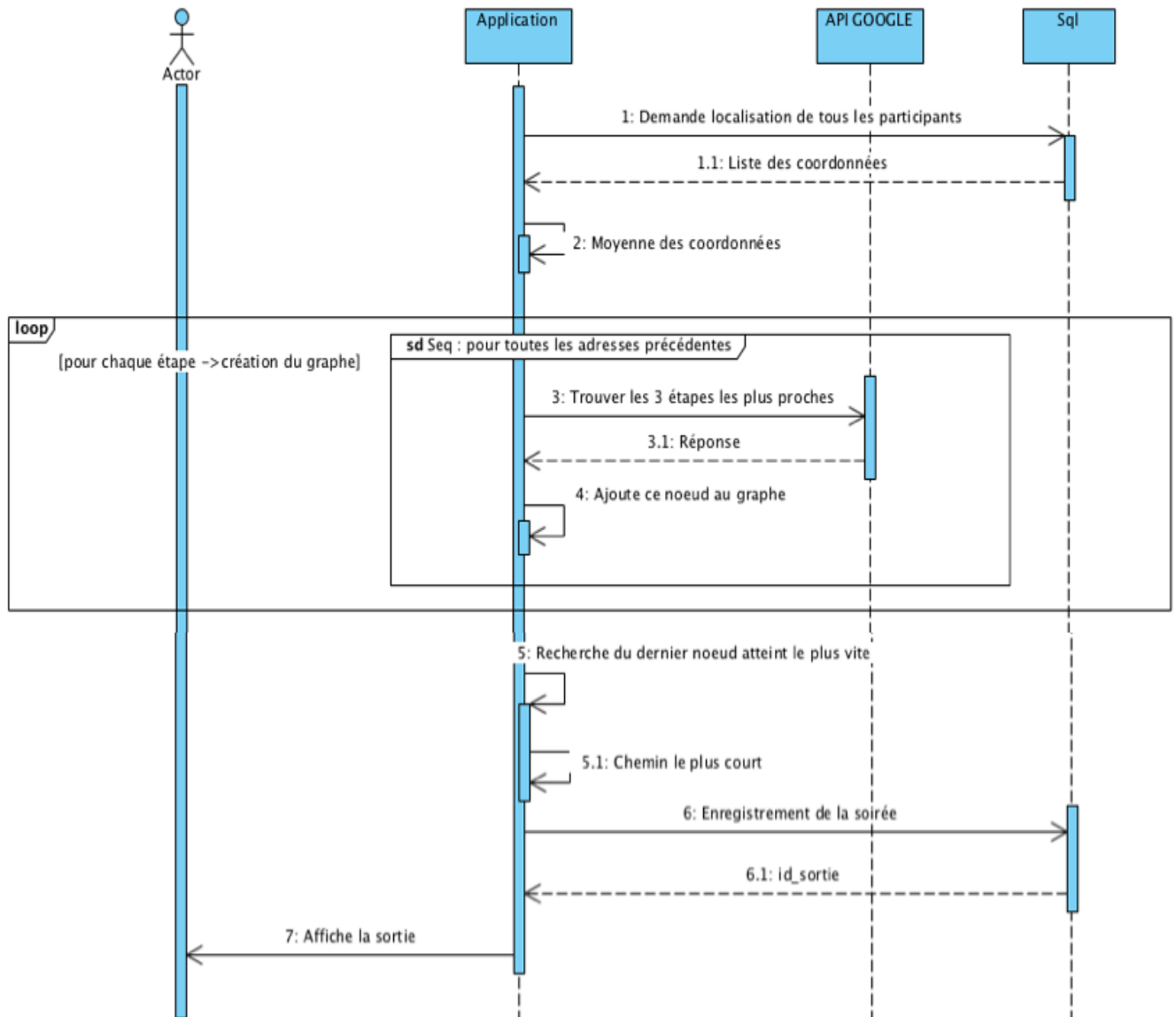


Figure 8 _ Diagramme de séquence représentant comment est calculé la sortie finale en fonction de tous les paramètres donnés précédemment

5. La carte

La carte exploitera une autre API de Google. Une barre de recherche sera présente ainsi que la possibilité de tapoter à un endroit de la carte pour l'ajouter aux favoris (comme spécifier ci-dessous).

Rechercher un lieu



Figure 9 – Interfaces représentant la carte

6. Tests

a. Tests unitaires

Tests	Objectifs	Données en entrée		Résultats attendus
T1	Valider sortie avec des données en entrée manquantes (nom ou date ou heure ou participants inexistants)	Nom lieu date heure participants Moyen de transport	Retrouvaille Paris Transport en commun	Message d'erreur suggérant à l'utilisateur de remplir les champs manquants
T2	Organiser une sortie en indiquant un lieu inexistant	Nom lieu date heure participants Moyen de transport	Promenade Cauchemar 23/03/2017 18:00 [Antoine,Emrique,florian] voiture	Message d'erreur indiquant que le lieu sélectionné n'est pas sur la carte
T3	Organiser une sortie dont la date est dépassée	Nom lieu date heure participants Moyen de transport	rencontre Orsay 23/03/2015 01:00 [Enzo,Thomas,Ibrahima] voiture	Message d'erreur indiquant que la sortie ne peut être créée à cette heure
T4	Créer une sortie au nom déjà existant	Nom lieu date heure participants Moyen de transport	Anniversaire Montcuq 23/03/2017 18:00 [Abbess,Cédric,Thierno] voiture	Message d'erreur indiquant que ce nom de sortie existe déjà
T5	Ajouter un participant sans indiquer son nom	nom adresse téléphone	22, rue conflans,95220 0623121214	Message d'erreur suggérant à l'utilisateur de saisir le nom
T6	Enlever la 2 ^{ème} étape	Type de lieu	Restaurant	Le numéro est effacé de Restaurant et la 3 ^{ème} étape se numérote 2
T7	Chercher un endroit sur la carte selon adresse	adresse	10 Avenue d'Iéna, 75116 Paris	Affiche sur la carte à l'aide d'un marqueur le point recherché

Tests	Objectifs	Données en entrée		Résultats attendus
T8	Chercher une adresse inexistante	adresse	12000 avenue la joie de travailler en Groupe	Afficher message d'erreur indiquant que l'adresse n'existe pas

b. Tests fonctionnels

Tests	Objectifs	Données en entrée		Résultats attendus
T1	Organiser une sortie	Nom lieu date heure participants Moyen de transport	Anniversaire Paris 23/03/2017 11:55 [Abbess, Cédric, Thierno] voiture	Passe à l'étape suivante
T2	Ajouter un participant à la sortie	nom adresse Email	Maguette NDAO 98, rue conflans, 95220 0612547890	le participant est ajouté à la sortie
T3	Affichage de la carte			La carte de la ville s'affiche grâce à Google Map sur notre page.

c. Tests d'acceptation

De manière très générale, il faudra que l'application reste fluide en toute condition, ou que les ralentissements soient compréhensibles (calcul en cours, chargement de la carte). De même, les bugs doivent générer des erreurs compréhensibles par l'utilisateur mais en aucun cas un plantage complet pour garantir la robustesse.

Cette application devra aussi respecter les contraintes qui ont été imposées par le cahier des charges.