



Projet Génie Logiciel Avancé L3

Cahier de conception

Assistant événementiel de sorties

CHOUKCHOU BRAHAM Abbes
NDIAYE Thierno Ismaila
VACHAUDEZ Cédric

Professeurs :
NGUYEN VAN Hai
FAISSOLE Florian

Année 2016-2017 L3 Informatique

Table des matières

1	Introduction	1
1.1	Caractéristiques du système visé	1
1.2	Dépendances	1
2	Structure globale	2
2.1	Base de données	4
3	Démarrage	5
4	Création d'une sortie	7
4.1	Initialisation de la sortie	7
4.2	Ajout de participants	9
4.3	Choix des étapes	10
4.4	Choix des contraintes	12
4.5	Calcul de la sortie idéale	12
5	Favoris	14
6	Carte	15
7	Tests	16
7.1	Tests unitaires	16
7.2	Tests fonctionnels	18
7.3	Tests d'acceptation	19

Table des figures

1	Diagramme états-transitions	2
2	Diagramme des classes principales	3
3	Diagramme entités-relations	4
4	Diagramme de séquence de l'accueil	6
6	Diagramme de séquence d'initialisation de sortie	8
9	Diagramme de séquence de modification des étapes	11
10	Diagramme de classe du graphe	12
11	Diagramme de séquence de calcul de la sortie	13
12	Diagramme de séquence d'utilisation des favoris	14

1 Introduction

Le but de ce document est de présenter l'architecture de l'application Best-ProjetGLA de manière plus détaillée que le cahier des charges ainsi que les tests qui devront être menés.

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 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

L'application devant fonctionner sur les téléphones Android en version 4.1.2 minimum, il faudra donc utiliser l'API 16. Le langage de programmation est très proche de Java et possède toutes ses caractéristiques, notamment la programmation orientée objet.

Les ressources sont très limitées. Il faudra donc consommer le moins de mémoire RAM et physique possible, limiter la demande en bande passante, et prendre en compte l'instabilité du réseau mobile 3G/4G. 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 et faire tourner l'application en mode déconnecté. Android implémentant depuis toujours SQLite, ce sera donc elle qui sera utilisée.

2 Structure globale

Les différentes actions possibles de l'utilisateur ont été représentées par le diagramme 1, tandis que les classes dont sera constituée l'application sont représentées dans le diagramme 2.

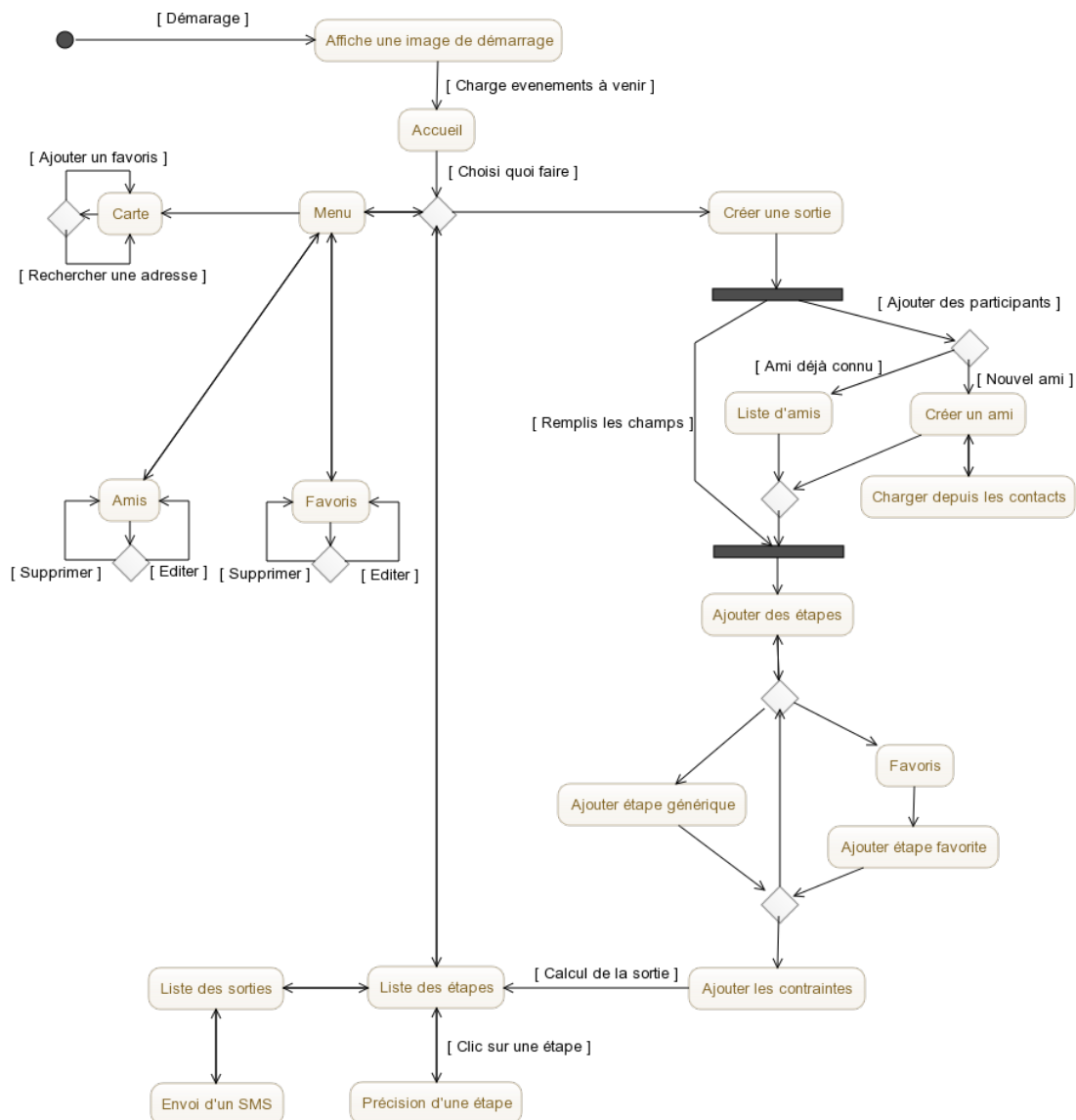


Figure 1 – Diagramme états-transitions représentant tous les déplacements possibles de l'utilisateur au sein de l'application

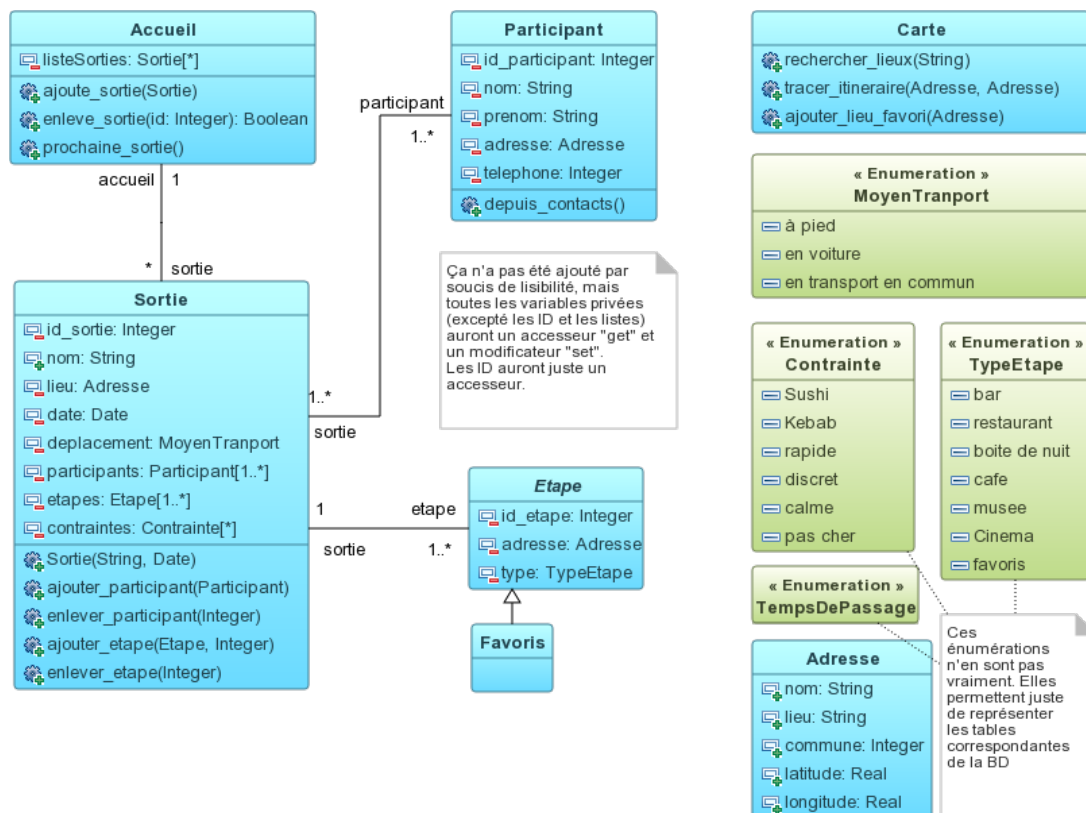


Figure 2 – Diagramme présentant les classes (et leurs relations) telles qu'elles seront implémentées dans notre application

2.1 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 favoris, 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.

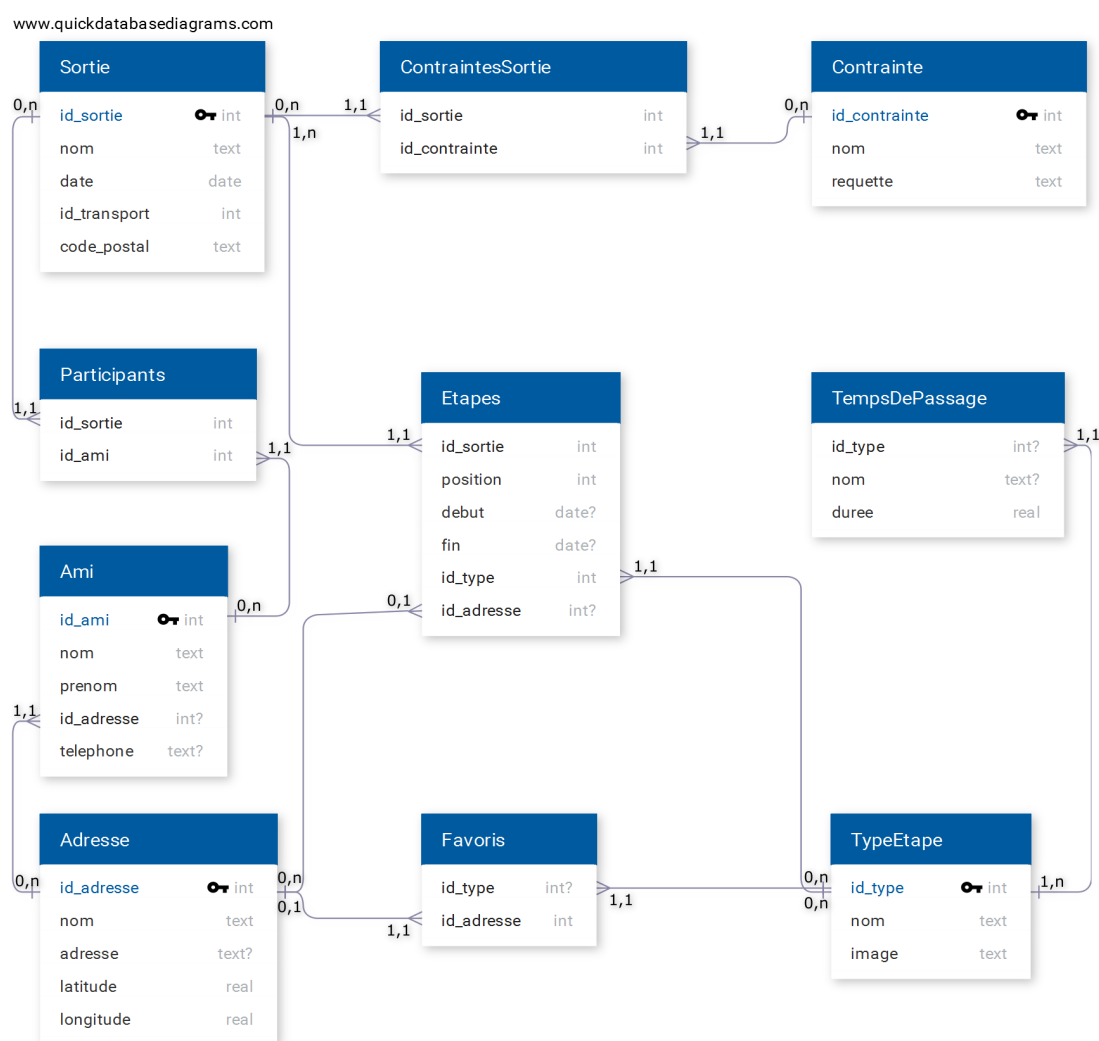
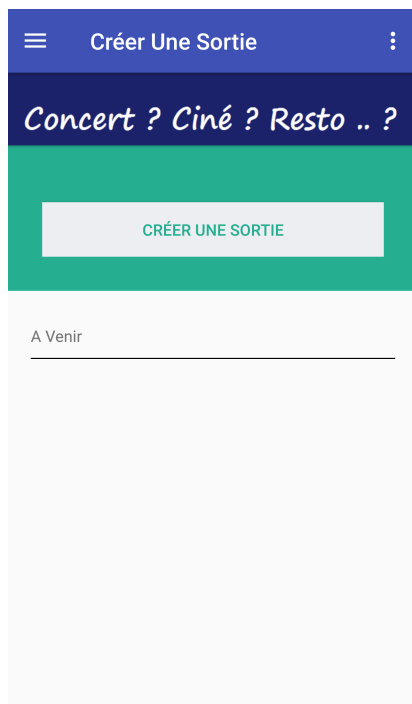


Figure 3 – Diagramme entités-relations présentant la structure de la base de données

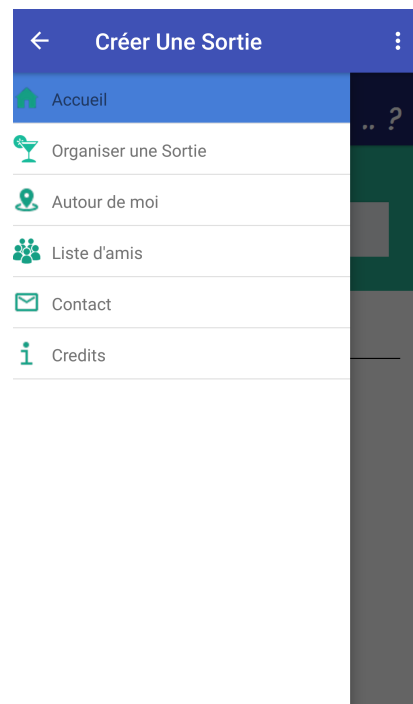
3 Démarrage

Le démarrage de l'application est assez sobre : une image s'affichera avec son nom et son logo, permettant de faire patienter l'utilisateur le temps qu'elle s'initialise et charge les futures sorties (s'il y en a) dans la base de données. Cette partie est présentée dans le diagramme 4.

On affichera ensuite l'accueil (interface 4a) qui donnera un aperçu des soirées à venir (ou en cours) ainsi qu'un accès rapide à toutes les fonctions de l'application à l'aide du menu (interface 4b).



(a) Accueil de l'application



(b) Menu de l'application

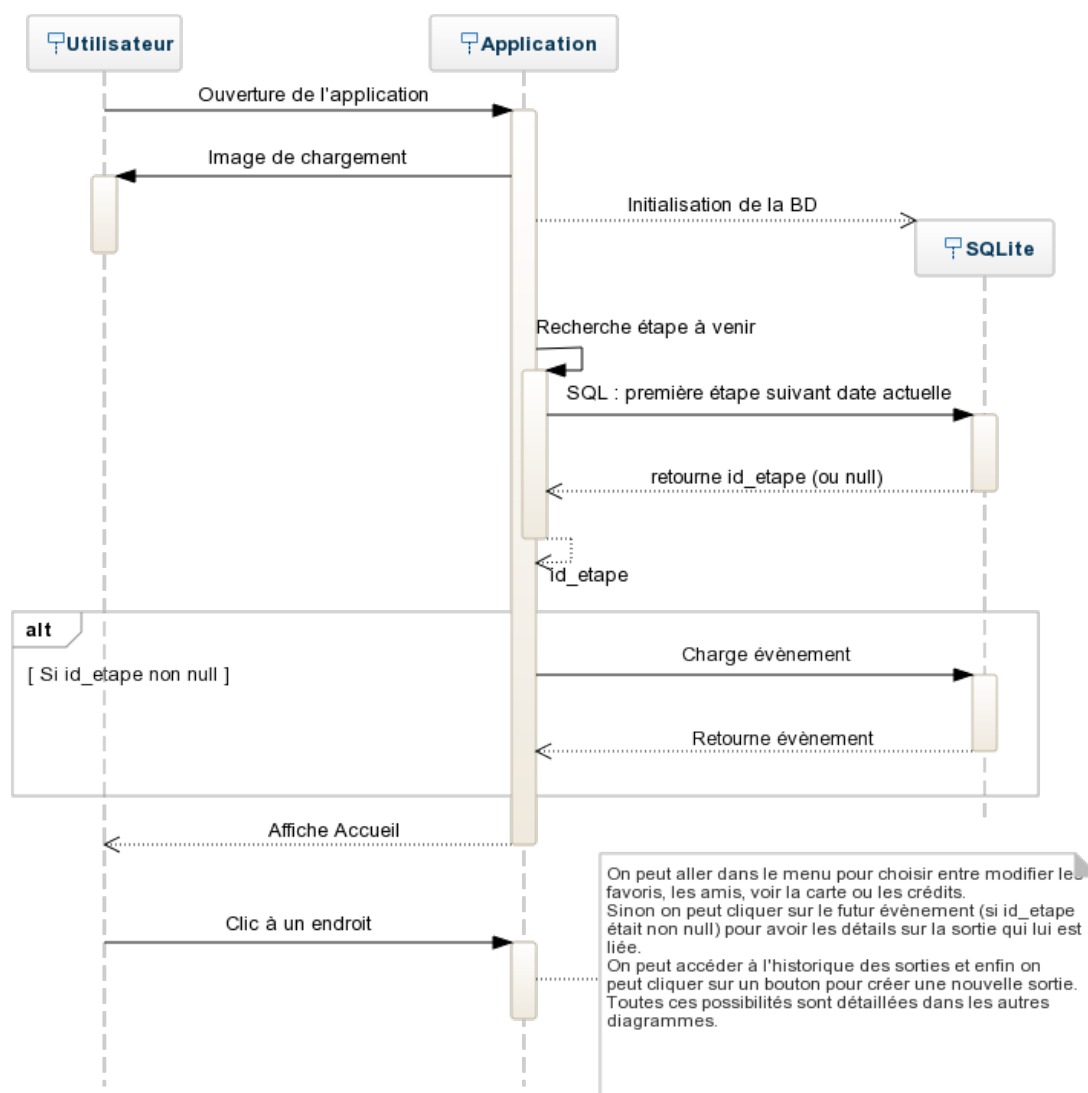


Figure 4 – Diagramme de séquence représentant le déroulement interne de l'application lors de son démarrage

4 Création d'une sortie

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

4.1 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 (interface 5) tout comme son fonctionnement (diagramme 6, ne demandant qu'à remplir les champs et cliquer sur son moyen de transport).

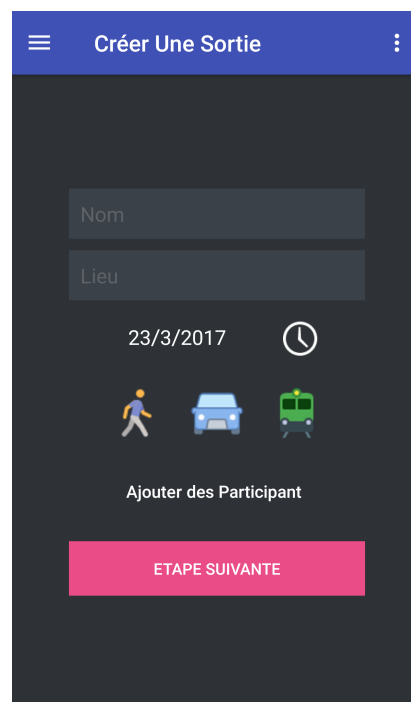


Figure 5 – Interface permettant d'initialiser une nouvelle sortie

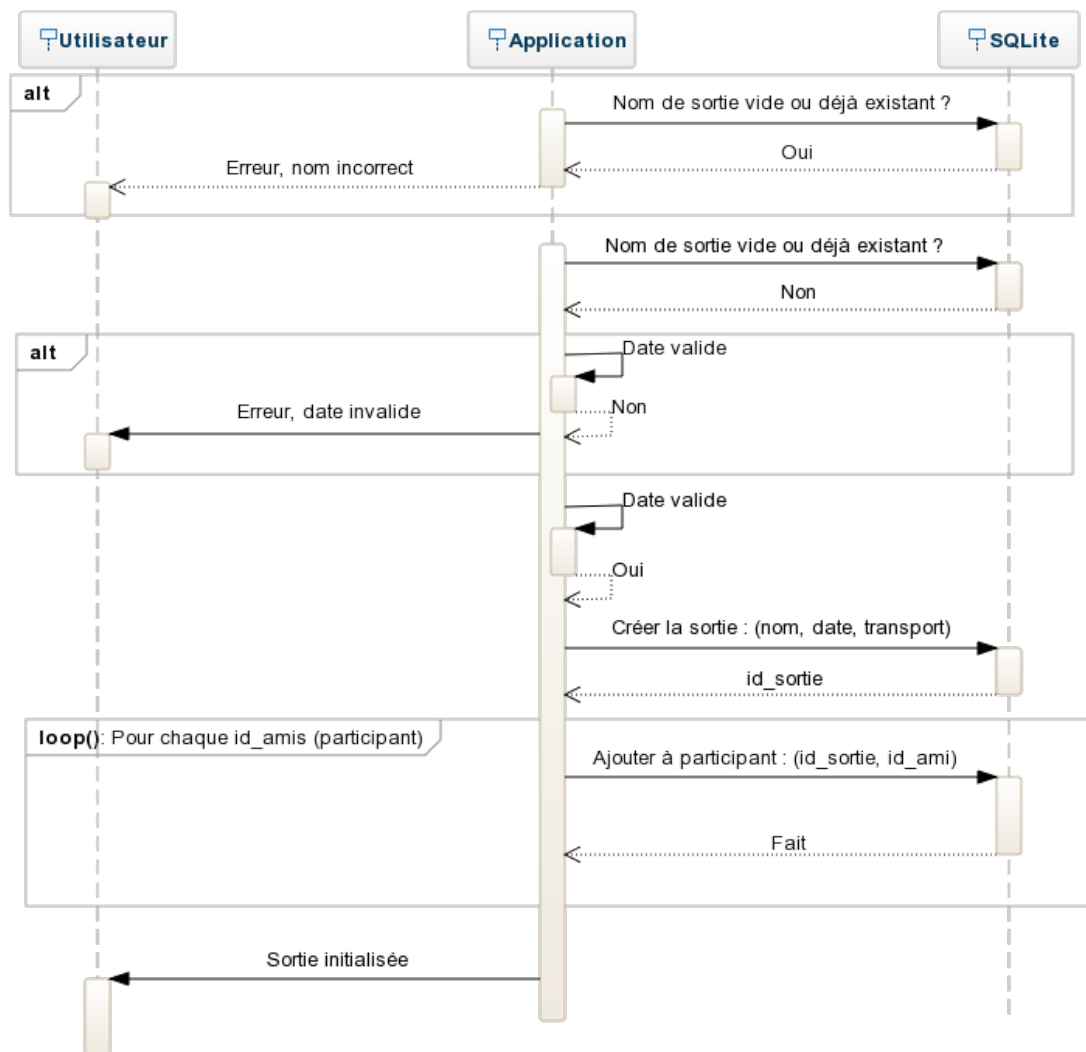
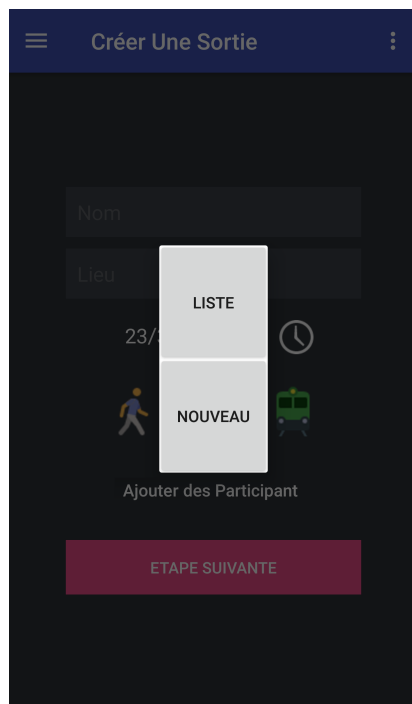


Figure 6 – Diagramme de séquence représentant les étapes et les cas d'erreur lors de l'initialisation d'une nouvelle sortie

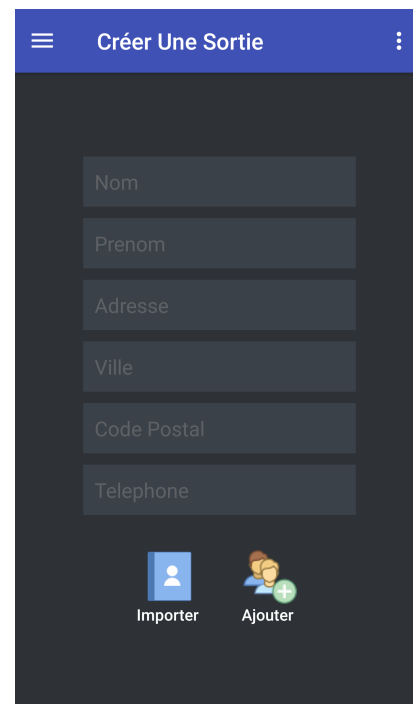
4.2 Ajout de participants

Chaque participant peut être ajouté de plusieurs manières différentes (interface 7a) : soit en le sélectionnant depuis la liste des amis (d'anciens participants), soit en en créant un nouveau (interface 7b). Un participant peut alors être importé depuis sa liste de contacts et complété si besoins est.

Si l'utilisateur vient de créer un participant, il sera ajouté en plus à sa liste d'amis.



(a) Choix entre créer un participant ou en sélectionner un ancien



(b) Création d'un nouveau participant

Figure 7 – Interfaces permettant d'ajouter des participants à la sortie

4.3 Choix des étapes

Vient ensuite l'étape permettant de définir les étapes de la sortie ainsi que leur ordre. Pour cela, l'interface 8a affichera les différents endroits possibles qu'il suffira de tapoter dans l'ordre voulu (exemple avec l'interface 8). Pour supprimer une étape il suffit de tapoter à nouveau dessus.

Tout cela est détaillé dans le diagramme 9.

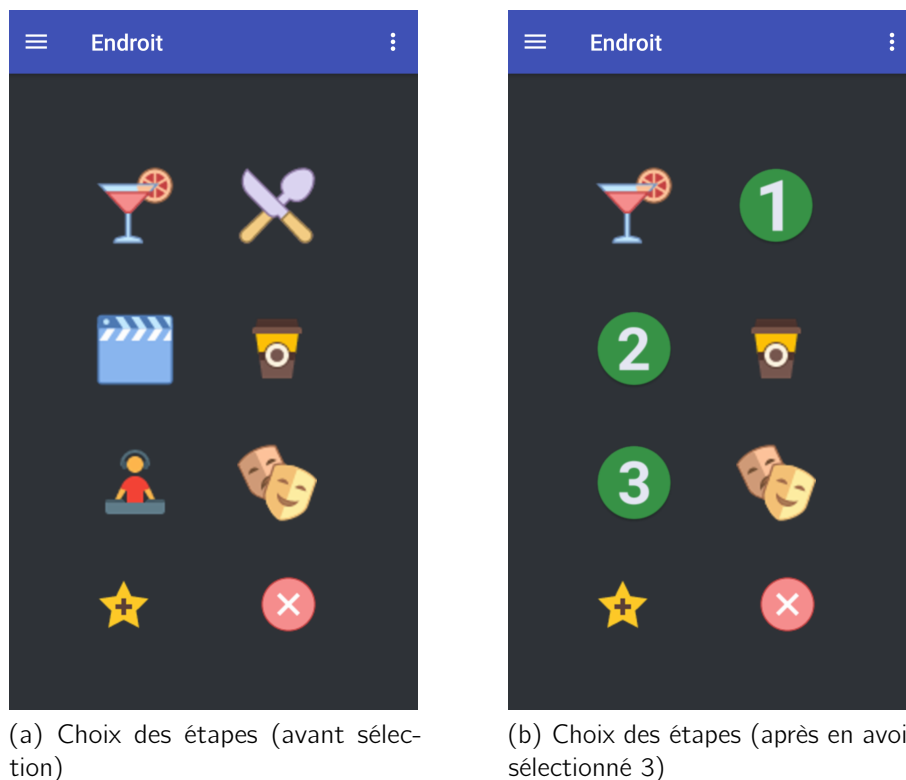


Figure 8 – Interfaces représentant le menu de choix des étapes, avant et après en avoir sélectionné

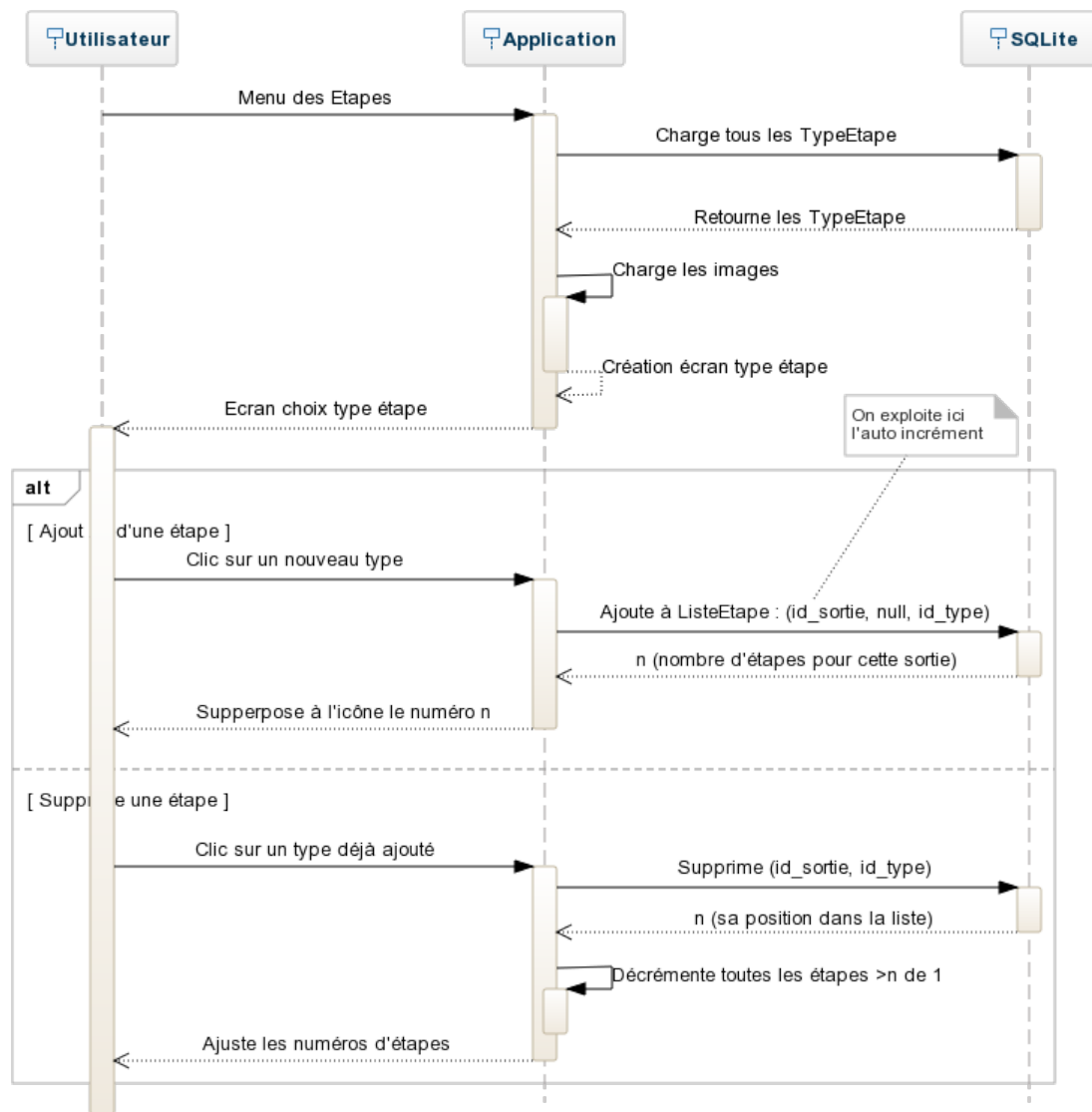


Figure 9 – Diagramme de séquence représentant comment ajouter ou enlever une étape à la sortie

4.4 Choix des contraintes

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

4.5 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 11 et consiste simplement en un algorithme de parcours en largeur d'un arbre. Cet arbre sera représenté à l'aide des classes 10, 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.

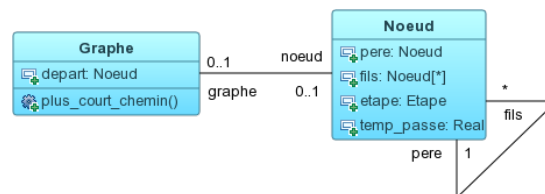


Figure 10 – Diagramme de classe utilisé pour représenter un graphe orienté d'étapes

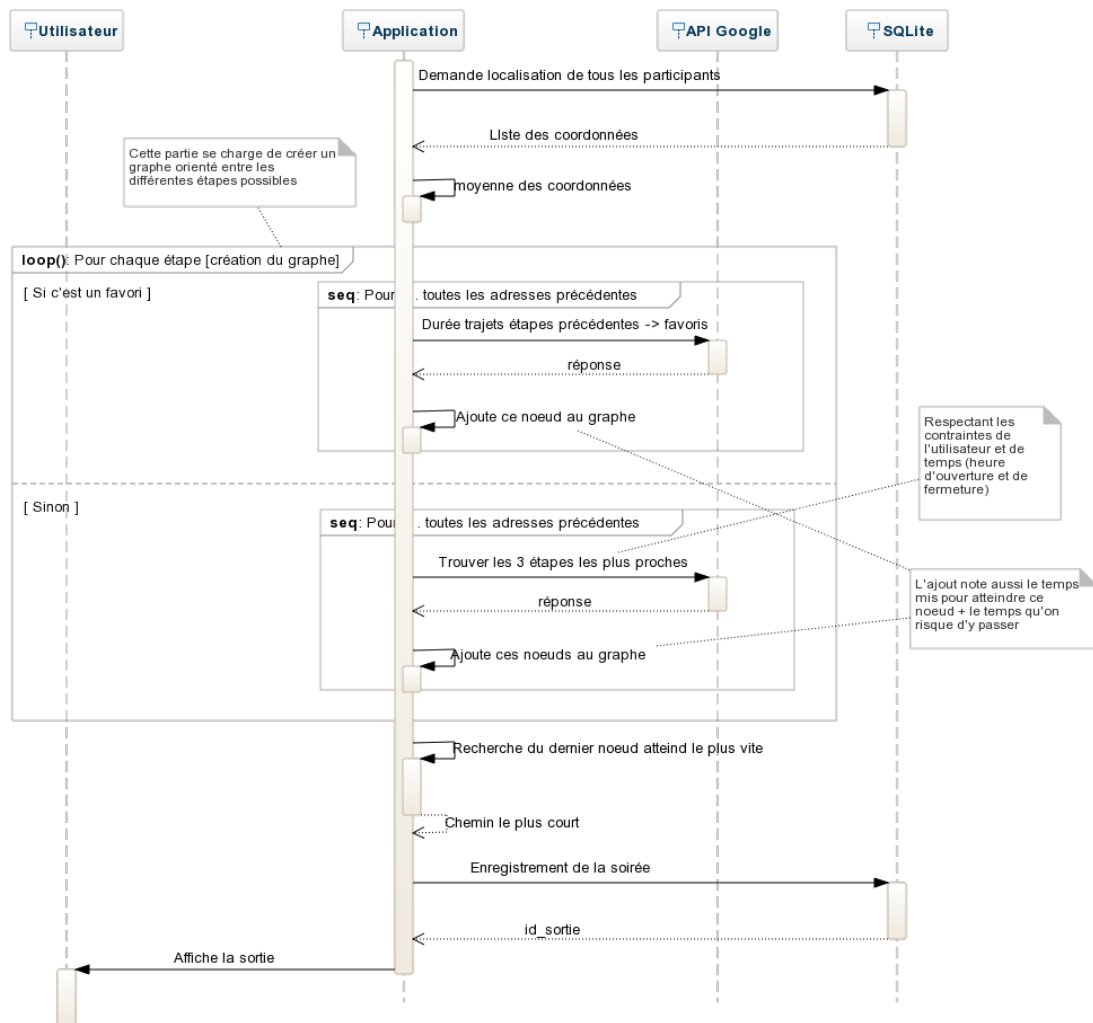


Figure 11 – Diagramme de séquence représentant comment est calculé la sortie finale en fonction de tous les paramètres donnés précédemment

5 Favoris

Le diagramme 12 décrit comment l'utilisateur procède pour ajouter une adresse à sa liste des favoris. Cela peut se faire en cherchant un lieu sur la carte ou par le biais d'un formulaire.

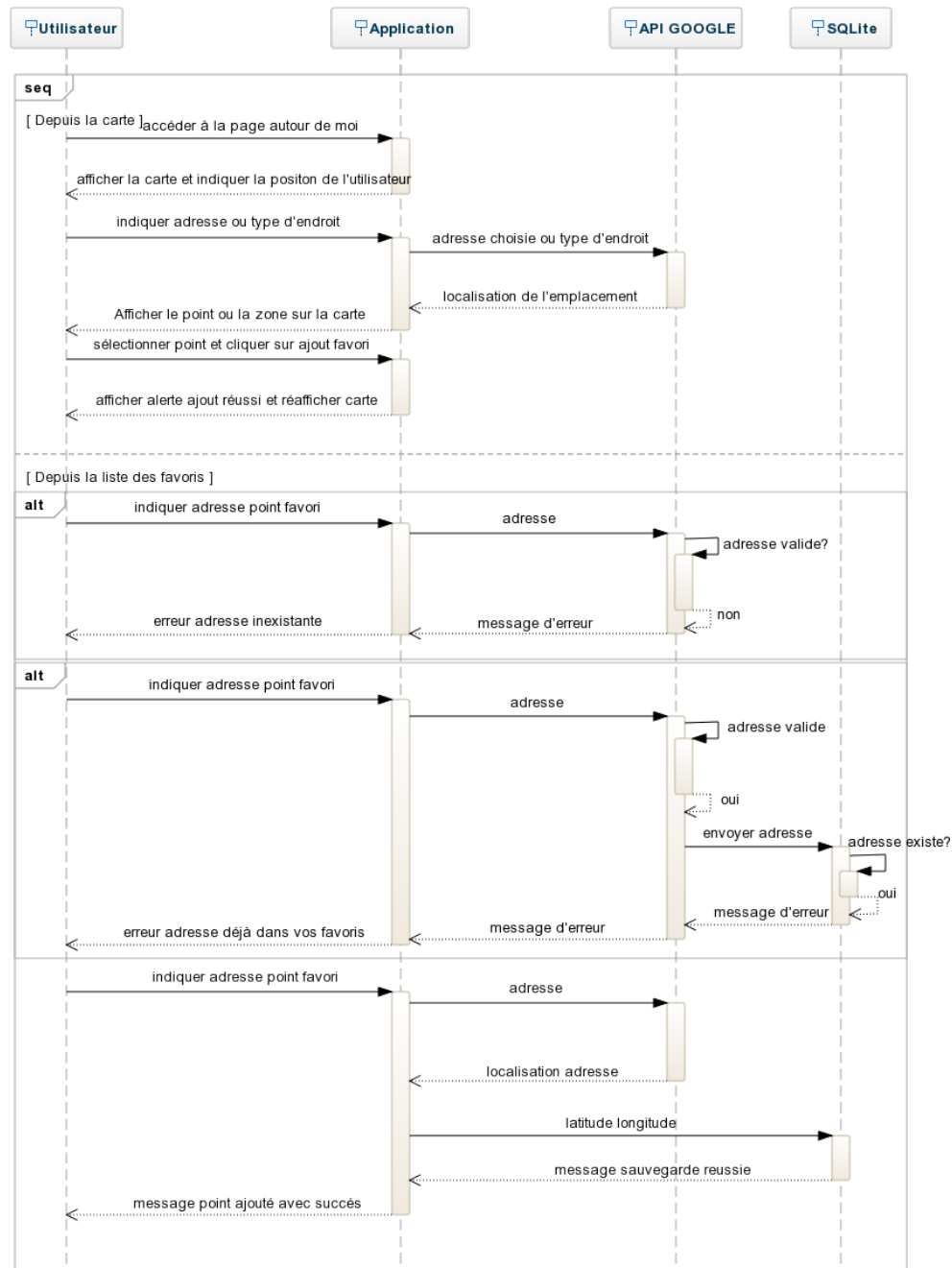


Figure 12 – Diagramme de séquence représentant comment modifier ses favoris (que ce soit l'ajout ou la suppression)

6 Carte

La carte demandera elle aussi une connexion internet pour fonctionner correctement et 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).



Figure 13 – Interfaces représentant la carte

7 Tests

7.1 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 numéro de tel	nom adresse téléphone	Yannick LOUBELO 12, rue conflans,95220	Le participant est ajouté comme nouvel ami dans la BD et comme participant dans la soirée
T6	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
T7	Enlever la 2 ^{ème} étape	Type de lieu	Restaurant	Le numéro est effacé de Restaurant et la 3 ^{ème} étape se numérote 2
T8	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
T9	Chercher une adresse inexistante	adresse	12000 avenue la joie de travailler en Groupe	Afficher message d'erreur indiquant que l'adresse n'existe pas
T10	Ajouter un point favori déjà existant	adresse	8 Rue Tournefort, 75005 Paris	Afficher message d'erreur indiquant que le lieu sélectionné est déjà dans favoris
T11	Lancer le calcul de la soirée avec internet et connexion réseau perdu avant l'affichage			Message d'erreur indiquant que la connexion réseau a été perdu
T12	Regarder la carte sans connexion internet			Message d'erreur indiquant qu'internet est nécessaire
T13	Commencer la création d'une sortie sans connexion internet			Fonctionne normalement
T14	Consulter la liste des sorties déjà organisées sans avoir créé			Liste vide
T15	Rechercher un nom inexistant parmi les Amis	nom	Uvuvuueucvuvueuvuevu u osaynsanynsdas ossas	Indique qu'aucun ami ne porte ce nom
T16	Rechercher un nom dans une liste d'amis vide			Indique que la liste d'amis est vide
T17	Cliquer sur une étape encore jamais ajouté			Place le numéro d'étape suivant par dessus

7.2 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 téléphone	Maguette NDAO 98, rue conflans, 95220 0612547890	Le participant est ajouté comme nouvel ami dans la BD et comme participant dans la soirée
T3	Ajouter une 3 ^{ème} étape	Type de lieu	Bar	Le numéro 3 apparaît sur l'icone Bar
T4	Affichage de la carte			La carte de la ville s'affiche grâce à Google Map sur notre page.
T5	Ajouter un point favori à partir de la carte	adresse	8 Rue Tournefort, 75005 Paris	Afficher sur la carte le nouveau point crée avec un marqueur
T6	Lancer le calcul de la soirée sans connexion internet			Message d'erreur indiquant qu'internet est nécessaire
T7	Rechercher un nom existant parmi les Amis	nom	Jean Jaque	Retourne toutes les personnes ayant « Jean Jaque » dans leur nom ou prénom
T8	Cliquer sur une étape déjà ajouté (numérotée)			L'étape n'est plus numérotée et toutes les suivantes sont décrémentées

7.3 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. Elle devra donc passer les tests suivants :

- Essayer de faire fonctionner l'application sans activer internet. Il doit être possible de visualiser les sorties enregistrées (passées ou à venir) et leurs étapes détaillées.
- Que la fonction d'envoi de SMS soit opérationnelle en permettant d'envoyer à tous les participants figurant dans la liste des contacts enregistrés, les informations sur la sortie sélectionnée.
- Que la création de sortie fonctionne de A à Z et que la sortie proposée soit valable en respectant les contraintes et les horaires des lieux de rendez-vous.
- Pourvoir supprimer des Amis et des Favoris de leurs listes respectives.