**Résumé :**

La migration est nécessaire lorsque des organisations ou des individus procèdent au changement de leurs systèmes informatiques ou à leur mise à niveau. Cette transformation vise à modifier l'ensemble des données gérées par un système informatique, source pour pouvoir les utiliser sur autre système cible ou bien pour séparer certaines informations des autres. La cohérence doit être maintenue entre l’ancien et le nouveau système après que les données aient été migrées. La migration des données est généralement programmée pour parvenir à un traitement automatisé, en libérant des ressources humaines des tâches embarrassantes. C’est dans ce contexte s’insère notre projet de fin d’études effectué au sein de la société AMARIS. Il consiste à la mise en place d’un produit qui offre la possibilité de faire le management événementiel tout en prenant en considération les différents intervenants dans ce domaine.

**Abstract** **:**

Migration is needed when organizations or individuals are changing or upgrading computer systems. This transformation aims at modifying all the data managed by a source computer system so as to be able to use them on another target system. Consistency must be maintained between the old and the new system after the data has been migrated. Data migration is usually programmed to achieve automated processing, freeing human resources from embarrassing tasks. It is in this context that our end of studies project carried out within the AMARIS company fits. It consists of the implementation of a product that offers the possibility of doing event management while taking into account the different players in this area.

**Dédicaces**

***A mes chers parents, ma fierté***

*Mon père, mon premier encadrant depuis ma naissance, Ma mère, qui a garni mes chemins,*

*Aucun mot, aucune dédicace ne peut exprimer mon immense respect, ce que je vous dois, et l’amour éternel pour les sacrifices que vous avez consentis pour mon instruction et mon bien être, sachez que je vous suis très reconnaissant et je ferai de mon mieux pour rester un sujet de fierté à vos yeux.*

***A ma sœur, mes amis***

*En témoignage de mes sincères reconnaissances pour leur aide et leur soutien moral durant l’élaboration du travail de fin d’étude, je vous souhaite une vie pleine de bonheur et de réussite.*

*Rami*

**Remerciements**

*Au terme de notre projet fin d’étude, je tiens à remercier toutes les personnes, qui par leurs conseils, leurs suggestions ou par leur simple présence nous ont permis de rendre notre travail aussi instructif et efficace que plaisant.*

*Je remercie tout spécialement notre encadreur Monsieur GARGOURI Karray ainsi que Monsieur BERRAYENA Seif pour leurs encadrements tout au long de ce projet, leur patience, leur disponibilité et leurs conseils toujours avisés.*

*Enfin, mes remerciements vont à tous les enseignants d’ESPRIT pour la formation qu'ils m’ont fournie et tous les membres du jury pour avoir accepté de juger ce travail.*

Table des matières

[Introduction Générale 1](#_Toc524628108)

[Chapitre 1 Contexte Générale 4](#_Toc524628109)

[Introduction 5](#_Toc524628110)

[I. Présentation générale 5](#_Toc524628111)

[1. Cadre du projet 5](#_Toc524628112)

[II. Présentation de l’organisme d’accueil 5](#_Toc524628113)

[1. Présentation Générale de la société 5](#_Toc524628114)

[2. Organigramme 6](#_Toc524628115)

[Chapitre 2 Etat de l’art 7](#_Toc524628116)

[I. L’art 8](#_Toc524628117)

[II. Etude de l’existant : Analyses et critiques 9](#_Toc524628118)

[1. L’existant 9](#_Toc524628119)

[2. Critique de l’existant 12](#_Toc524628120)

[III. Présentation du projet 13](#_Toc524628121)

[1. Problématique 13](#_Toc524628122)

[2. Objectif du projet 14](#_Toc524628123)

[IV. Solution envisagée 15](#_Toc524628124)

[V. Choix de la méthodologie 16](#_Toc524628125)

[1. Etude Comparatif 16](#_Toc524628126)

[2. SCRUM 19](#_Toc524628127)

[3. Pourquoi SCRUM ? 19](#_Toc524628128)

[Chapitre 3 Analyse et spécification des besoins 21](#_Toc524628129)

[Introduction 22](#_Toc524628130)

[I. Identification des acteurs 22](#_Toc524628131)

[II. Identification des besoins 23](#_Toc524628132)

[1. Besoins fonctionnels 23](#_Toc524628133)

[*2.* *Besoins non fonctionnels* 23](#_Toc524628134)

[III. Diagramme de cas d’utilisation 24](#_Toc524628135)

[IV. Scénario des cas d’utilisation 25](#_Toc524628136)

[V. Diagramme de classe Analyse 28](#_Toc524628137)

[VI. Diagramme de séquence Système : 29](#_Toc524628138)

[VII. Maquettes 30](#_Toc524628139)

[VIII. Backlog du produit 31](#_Toc524628140)

[IX. Architecture Physique 33](#_Toc524628141)

[X. Architecture Logique 34](#_Toc524628142)

[XI. Base de données 35](#_Toc524628143)

[XII. Diagramme de déploiement 36](#_Toc524628144)

[Chapitre 4 Conception 38](#_Toc524628145)

[Introduction 39](#_Toc524628146)

[I. Conception Préliminaire 39](#_Toc524628147)

[1. Les Patrons de conception 39](#_Toc524628148)

[II. Conception détaillé 41](#_Toc524628149)

[1. Diagramme d’entité 41](#_Toc524628150)

[III. Diagramme de séquence 42](#_Toc524628151)

[1. Diagramme de séquence Objet (Assign Owner) : 42](#_Toc524628152)

[2. Diagramme de séquence Objet (Assign Membre) : 44](#_Toc524628153)

[3. Diagramme de séquence Objet (Request Event) : 45](#_Toc524628154)

[IV. Diagramme d’état transition 46](#_Toc524628155)

[Chapitre 5 Réalisation 47](#_Toc524628156)

[Introduction 48](#_Toc524628157)

[I. Environnement du travail 48](#_Toc524628158)

[1. Environnement logiciel 48](#_Toc524628159)

[2. Langages utilisés 48](#_Toc524628160)

[II. Scénarios 51](#_Toc524628161)

[1. Scénario Request Evénement 51](#_Toc524628162)

[2. Scénario Reject Evénement 53](#_Toc524628163)

[3. Scénario Allocation Owner pour un événement 55](#_Toc524628164)

[4. Scénario Allocation BookingAgent et Négociator pour un Evénement 55](#_Toc524628165)

[5. Scénario invitation des participants 58](#_Toc524628166)

[Annexe 61](#_Toc524628167)

[I. La plateforme Android 61](#_Toc524628168)

[1. Présentation 61](#_Toc524628169)

[2. Architecture de l’Android 61](#_Toc524628170)

[Conclusion générale 64](#_Toc524628171)

[Webographie 68](#_Toc524628172)

Liste des figures

[Figure 1 Logo Amaris 5](#_Toc524628173)

[Figure 2 Organigramme Amaris 6](#_Toc524628174)

[Figure 3 Interface EventBrite 9](#_Toc524628175)

[Figure 4 Exemple de Trello et son fonctionnement 10](#_Toc524628176)

[Figure 5 Exemple CVENT Event Management 11](#_Toc524628177)

[Figure 6 Processus SCRUM 19](#_Toc524628178)

[Figure 7 Diagramme de cas d'utilisation général 24](#_Toc524628179)

[Figure 8 Diagramme de classe d'analyse 28](#_Toc524628180)

[Figure 9 Diagramme séquence Systéme 29](#_Toc524628181)

[Figure 10 Maquette liste des proposal 30](#_Toc524628182)

[Figure 11 Maquette Event Brief 30](#_Toc524628183)

[Figure 12 Backlog du produit LifeStyle sur TFS 31](#_Toc524628184)

[Figure 13 Architecture Physique 33](#_Toc524628185)

[Figure 14 Architecture logique 35](#_Toc524628186)

[Figure 15 Diagramme de déploiement 36](#_Toc524628187)

[Figure 16 Architecture MVC 40](#_Toc524628188)

[Figure 17 Diagramme de classe de conception 41](#_Toc524628189)

[Figure 18 Diagramme séquence objet (Owner) 42](#_Toc524628190)

[Figure 19 Diagramme séquence objet (Membre) 44](#_Toc524628191)

[Figure 20 Diagramme séquence objet Requestor 45](#_Toc524628192)

[Figure 21 Diagramme d'état transition 46](#_Toc524628193)

[Figure 22 Utilisateur non autorisée a utilisé l'application 51](#_Toc524628194)

[Figure 23 Utilisateur autorisé à faire une demande uniquement 52](#_Toc524628195)

[Figure 24 Interface d'accueil 52](#_Toc524628196)

[Figure 25 Demande d'un événement 53](#_Toc524628197)

[Figure 26 Interface de rejet d'un événement 53](#_Toc524628198)

[Figure 27 Interface des détails d'un événement 54](#_Toc524628199)

[Figure 28 Interface du motifs de rejet d'un événement 54](#_Toc524628200)

[Figure 29 Interface d'assignation d'un Owner 55](#_Toc524628201)

[Figure 30 chronologie des événement 56](#_Toc524628202)

[Figure 31 Assignation des différents membres (BookingAgent/Négociator) 57](file:///C:\Users\RLANDOLSI\Documents\Rapport\RapportFinaleVBeta.docx#_Toc524628203)

[Figure 32 Interface du choix d'assignation 57](#_Toc524628204)

[Figure 33 Interface d'invitation des participants 58](#_Toc524628205)

[Figure 34 Interface des listes des participants 59](#_Toc524628206)

[Figure 35Architecture Android 61](#_Toc524628207)

Liste des tableaux

[Tableau 1: Tableau comparative entre les méthodes classiques et les méthodes agiles 18](#_Toc524628208)

[Tableau 2 Description détaillé sur la demande d'évènement. 25](#_Toc524628209)

[Tableau 3 Description détaillé sur l'allocation d'un Owner. 25](#_Toc524628210)

[Tableau 4 Description détaillé sur l'assignation de l'équipe évent. 26](#_Toc524628211)

[Tableau 5 Description détaillé sur la recherche de proposition 26](#_Toc524628212)

[Tableau 6 Description détaillé sur la négociation. 27](#_Toc524628213)

[Tableau 7 Description détaillé sur la Team Member. 27](#_Toc524628214)

[Tableau 8 Exemple Backlog (Lifestyle) 32](#_Toc524628215)

# Introduction Générale

e nos jours la technologie évolue à un rythme excessif et surtout dans le monde de l’informatique. Ces dernières années la technologie a connu un essor et une croissance exponentielle pour répondre à toutes les exigences et cela dans divers domaines tels que l’événementiel.

D

Ces progrès technologiques ouvrent de nouvelles perspectives commerciales qui donnent naissance à de nouveaux services, méthodes de travail et outils de gestion permettant une parfaite mise en œuvre des ressources de l’entreprise afin d’atteindre leurs objectifs pouvant être sociaux et/ou financier.

L'univers entrepreneurial a toujours eu une relation profonde avec l'informatique et les outils bureautique que cette technologie génère. L'un des exemples le plus probant est l'omniprésence de l'ERP (Entreprise Ressource Planning) dans la quasi-totalité des entreprises de ce monde. Ce type de logiciel est un gestionnaire de ressources dans le milieu entrepreneurial. Il permet de gérer les ressources d'une société à travers un système informatique local. Aujourd'hui, Plusieurs secteur de l'informatique bénéficient d'une addition significative d'efficacité dans leur action, nous allons voir l’utilité de l’ERP.

Le présent document s’articule autour de cinq chapitres et illustre le travail que nous avons accompli tout au long de notre stage.

Commençons par le premier chapitre, il s’intitule « Contexte générale », dans lequel nous allons présenter la société d’accueil, ses activités ainsi que son organigramme.

Pour le second chapitre, « Etat de l’art », nous étudierons l’art, nous définirons notre problématique ainsi que notre besoin, les solutions existantes, que nous critiquerons par la suite et nous justifierons finalement notre choix de la méthodologie que nous adapterons pour ce projet.

Le troisième chapitre, quant-a lui, il contiendra l’analyse et la spécification des besoins, où nous élaborerons les différentes fonctionnalités que nous intégrerons.

Nous entamerons par la suite le 4ème chapitre, intitulé « Conception » dont lequel nous révélerons plusieurs détails de conception grâce aux diagrammes de cas d’utilisations, séquences ainsi que ceux de classes.

Nous finirons par le 5ème et dernier chapitre, celui de la réalisation, où nous nous intéresserons aux environnements de travail qu’ils soient logiciels, matériels ainsi qu’à la réalisation et aux interfaces constituant l’application développée.

# Chapitre 1 Contexte Générale

## Introduction

Dans ce premier chapitre nous vous présenterons l’organisme d’accueil, les motivations ainsi que le contexte du projet à réaliser. Nous finirons par présenter un organigramme de l’organisme d’accueil.

## Présentation générale

### Cadre du projet

Ce projet est né dans le cadre de la formation d’un ingénieur informatique où on est amené à la fin de ce cursus à réaliser un rapport pour un projet de fin d’études en vue de l’obtention du diplôme national d’ingénierie de l’école supérieur privé de l’informatique et de la technologie (ESPRIT).

## Présentation de l’organisme d’accueil

### Présentation Générale de la société

Amaris est un cabinet de consulting opérant essentiellement dans le domaine du télécom.

Fondée en 2007, Amaris a su par la suite diversifier ces activités et repérer les besoins ce qui fait qu’aujourd’hui, qu’elle touche à plusieurs secteurs tels que le télécom, l’IT, le pétrole, le nucléaire, l’évènementielle, le voyage, la conciergerie, la finance, le domaine pharmaceutique, l’aérospatial, l’aéronautique, etc. avec un chiffre d’affaire avoisinant les 260 M€ et près de 5000 employées.

Aujourd’hui, Amaris est devenu un groupe qui ne cesse d’acquérir des sociétés stratégiques et dans des multiples domaines. Ce qui la caractérise c’est son implémentation à l’échelle mondial, en effet elle se localise dans plus de 90 pays.



Figure 1 Logo Amaris

### Organigramme

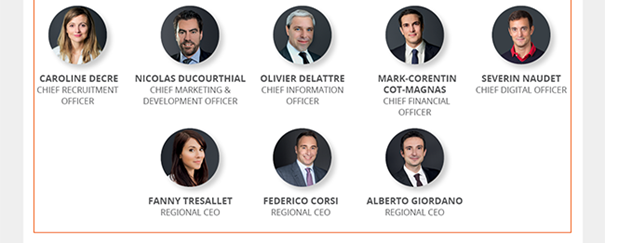
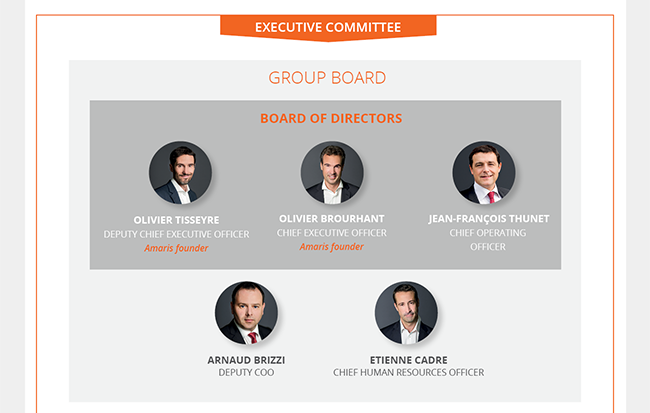


Figure 2 Organigramme Amaris

# Chapitre 2 Etat de l’art

## L’art

L’art est une activité humaine, le produit de cette activité ou l'idée que l'on s'en fait est le fruit de l’être humain, de ce fait dans notre cas qu’est ce que l’art ?

Imaginons, que le fait de créer un événement culturel engendre automatiquement le contact d’un traiteur pour les apéritifs de la cérémonie, du fleuriste, du décorateur, la négociation et l’acquisition du lieu, la génération des invitations et leur envoie cela nous faciliterait énormément les tâches.

Imaginons, maintenant qu’on peut superviser l’ensemble de ces tâches, cela semblerait fascinant.

De ce fait, on peut définir l’art propre à nous, vu que c’est le fruit de l’être humain définit précédemment ; l’art ici est la logistique.

Cependant, il existe plusieurs autres définitions de l’art, pour nôtre projet nous allons nous intéresser uniquement à la logistique.

Venant maintenant à la définition de la logistique ; La logistique en entreprise est d’autant plus une fonction transversale aux autres services que stratégique et influence considérablement l'activité de l'entreprise. Elle constitue un service à part dans la majorité des grandes et moyenne entreprises. Il s’agit donc d’optimiser la gestion de leurs moyens et ressources afin d’atteindre leurs objectifs.

Pour mieux comprendre la logistique, prenant un exemple au sein d’Amaris ; la création d’événement au sein du groupe fait mine d’une grande malice de la logistique ; En effet dès qu’un employé fait la demande d’un événement qui va se dérouler dans un autre pays, cet événement nécessite le déplacement de quelques autres employés, il procède automatiquement à la réservation d’un billet d’avion, la réservation d’une chambre d’hôtel, le taxi pour aller et revenir de l’aéroport, rajoutant à cela les négociations pour avoir un emplacement agréable, les décorateurs, la nourriture, tout ce processus définit un exemple de logistique interne du groupe.

## Etude de l’existant : Analyses et critiques

Actuellement, Amaris essaie de perfectionner son ERP et ne cesse de l’enrichir avec des nouvelles applications et fonctionnalités. Tels que le produit Lifestyle qui est composé en quatre partie à savoir :

* EventManagement
* BigEventManagement
* TravelAgency
* Conciergerie

### L’existant

Nous allons nous focalisées sur l’EventManagment et le TaskManagement ; à ce niveau il existe plusieurs autres applications que nous allons découvrir :

**A close up of a sign

Description generated with high confidence**

Application dédiée à la découverte des événements culturels et non à la gestion des événements. En effet celle-ci permet de consulter la liste des événements qui se déroulent et se dérouleront prochainement, mise à part, elle permet aussi l’achat de tickets en ligne.

Nous trouverons plus de détails dans l’interface ci-dessous où elle illustre bien des exemples d’événements qui vont se dérouler à Tunis.

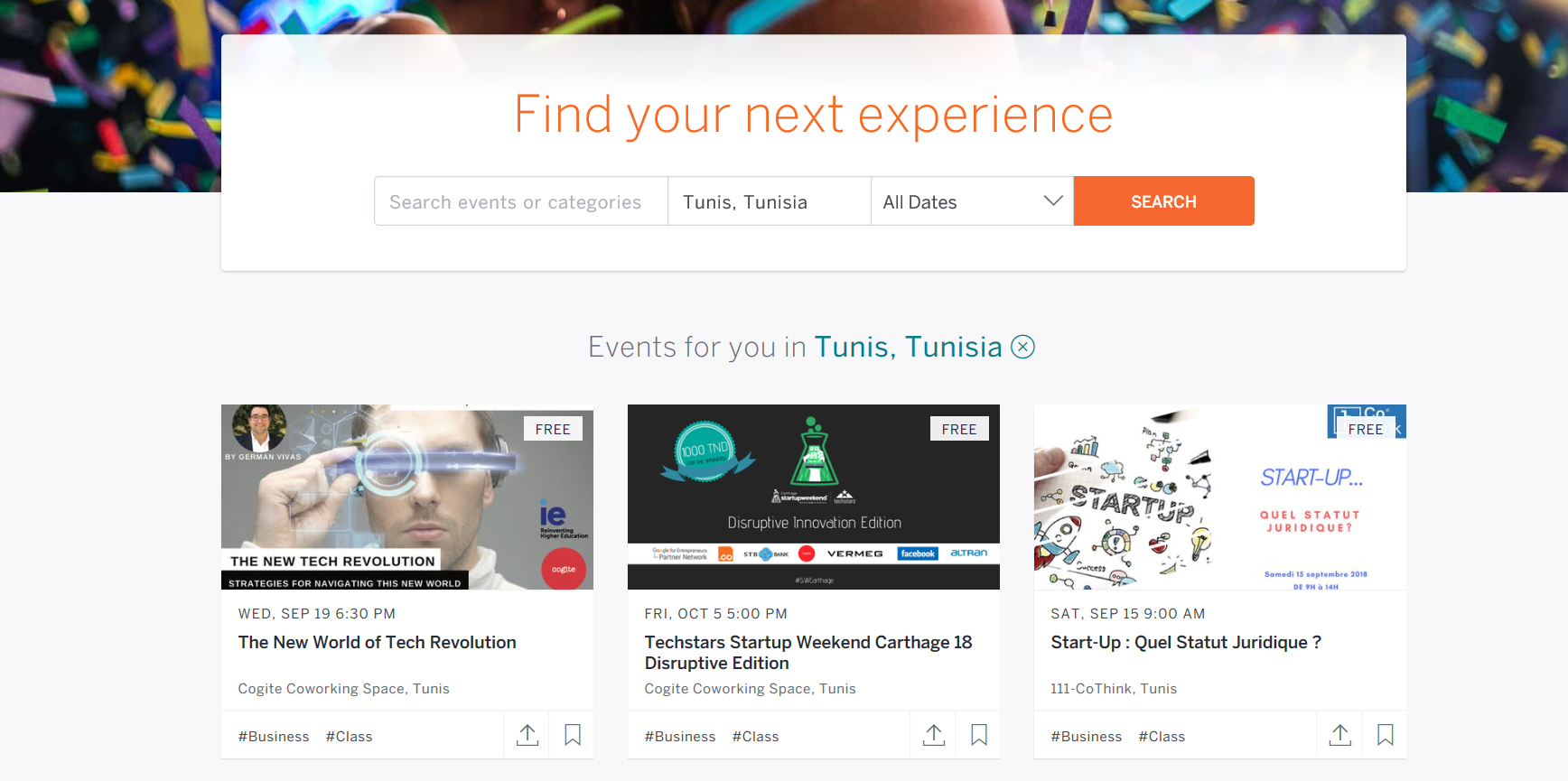


Figure 3 Interface EventBrite



Application dédiée à l’organisation des tâches, fondée en 2011 cette dernière permet d’avoir un visuel sur l’ensemble des tâches, l’état de leurs avancements et sur qui fait quoi. Il est basé sur la méthodologie Kanban de Toyota.

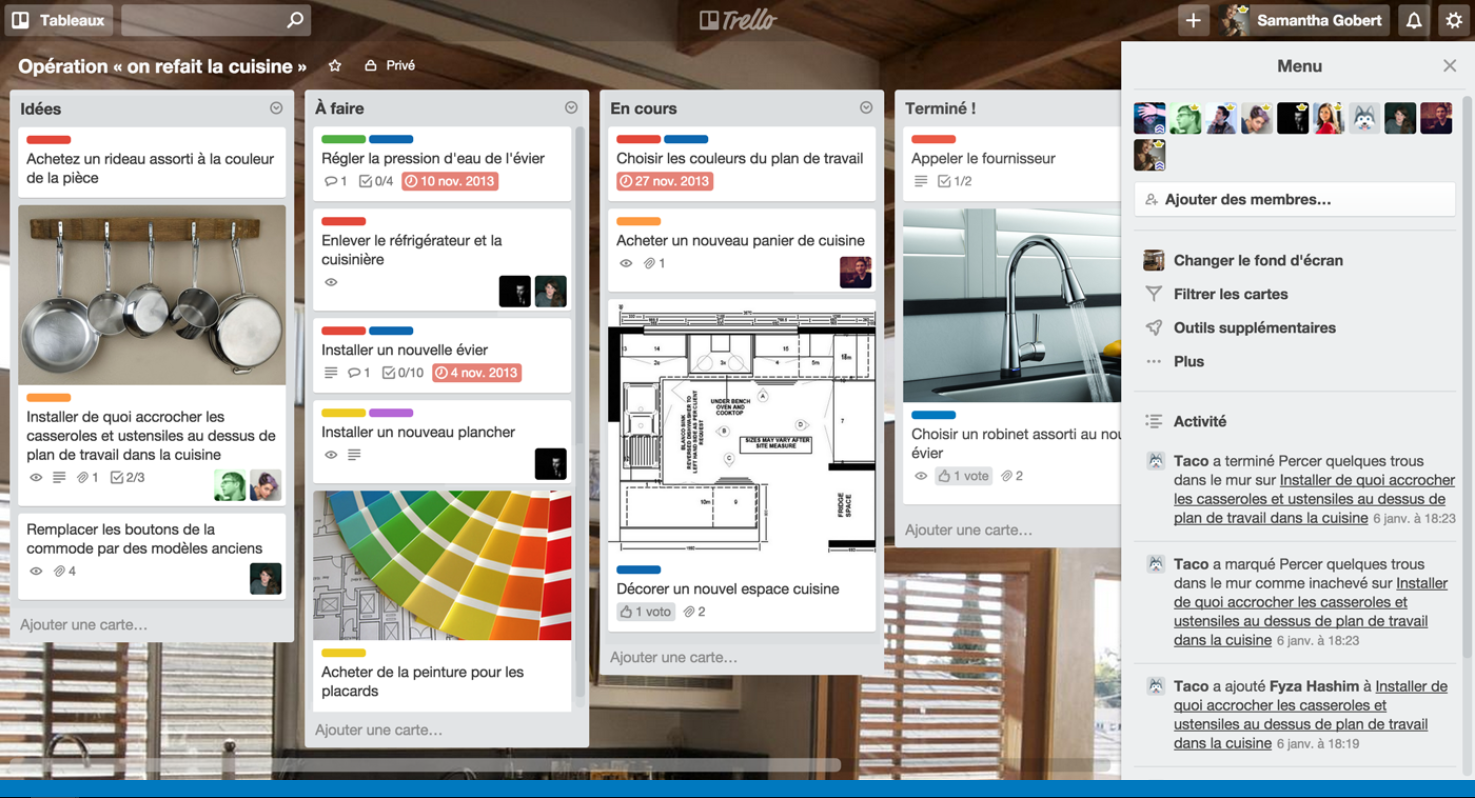
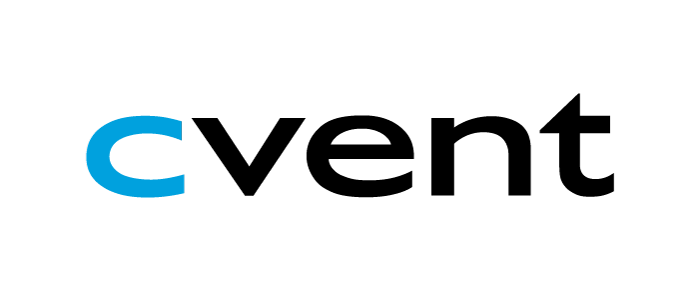


Figure 4 Exemple de Trello et son fonctionnement

****

Cette plateforme est un CRM de gestion des événements selon leurs types à savoir : réservation d’hôtel, avion, meeting, conférence, …

Fondée en 1999, cette plateforme Américaine à aujourd’hui plus de 250.000 fournisseur et partenaire.

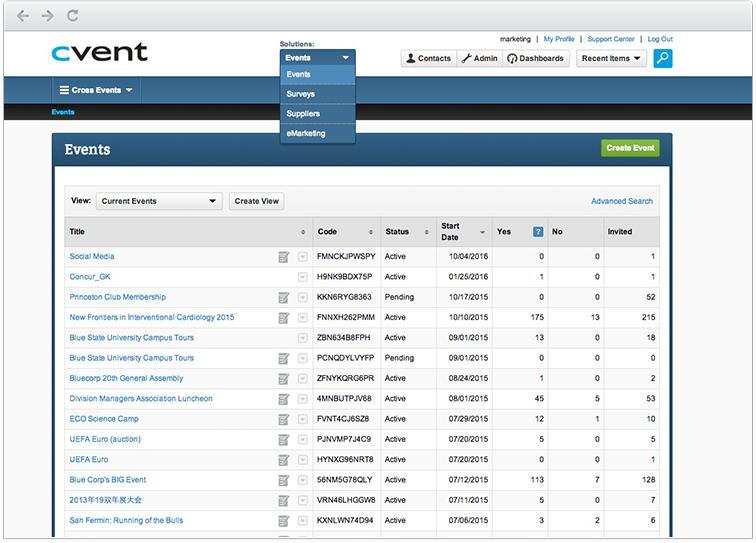


Figure 5 Exemple CVENT Event Management

Cette interface bien qu’elle soit simple elle présente un exemple pour le management d’un événement avec différentes fonctionnalités tels que le Survey.



Est une agence de gestion et d’organisation d’événement Canadienne. Anciennement appelé 3R elle devient par la suite EVENTUO.

### Critique de l’existant



EVENTUO n’est pas une solution informatisée ni automatisée, elle ne procure aucun visuel sur l’ensemble des tâches à effectuer ni effectuées, aucune visibilité sur qui fait quoi et quand. Rajoutant à cela qu’une intervention humaine et obligatoire.

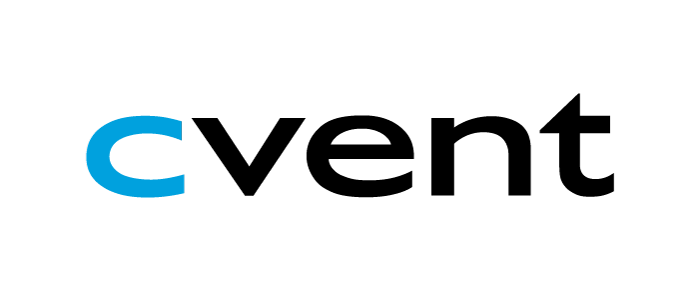


Trello est une solution pour la gestion des tâches si on peut le résumer ainsi, donc selon nôtre problématique et notre besoin elle résout certes quelques une d’entre elles mais ne résout pas la majorité de nos problématiques.

A close up of a sign

Description generated with high confidence

Certes cette solution à proposer une meilleure visibilité sur le monde des événements mais elle ne présente qu’une fonctionnalité parmi d’autres. Donc peut-on conclure qu’elle ne répond que partiellement à nôtre besoin.



Cette solution se rapproche le plus de nôtre besoin. En effet cette solution procure l’ensemble des fonctionnalités dont on a besoin à un détail près et c’est ce qui fait la différence.

En effet CVENT ne procure pas de liaison entre les différents types d’événements c’est-à-dire que par exemple pour un événement « Travel » qui nécessite l’achat un billet d’avion, la réservation d’hôtel, le taxi pour le déplacement d’un client ou un employé nous avons besoin de créer plusieurs processus à savoir : un pour l’achat d’un billet d’avion, un pour la réservation d’hôtel, un autre pour la gestion de l’événement tandis que nôtre besoin est de créer l’ensemble de ces tâches en même temps et sur une même application.

## Présentation du projet

### Problématique

Dans un groupe qui emploie plus que 5000 employés, qui ne cesse de croitre spécialement en Tunisie et qui compte aujourd’hui plus que 120 employées dont près de 60 consultants en télécom. Ces consultants ne cessent de voyager, recevoir leurs clients, les inviter aux restaurants, les accueillir à l’aéroport, les ramener à leurs hôtels. Toute cette gestion ainsi que celle de ses projets, ses employés et ses ressources deviennent des tâches assez fastidieuses.

Que faire lorsque on reçoit une demande d’un événement, comment différencier les types de ces derniers, par où commencer ?

Tout au long des différents processus, le responsable rencontre beaucoup de difficultés pour la planification et le suivi de ses tâches.

En effet, dès qu’il reçoit une demande d’événement celui-ci commence par valider l’évènement avec son supérieur selon son budget et sa description par la suite il lui affecte un Owner qui peut-être non disponible. Par la suite l’Owner affecte des membres pour l’organisation de cet évènement à savoir un Booking Agent, un Negociator, des membres sans aucun rôle et ces derniers peuvent être non disponible lors du déroulement de ce dernier. Par la suite il entame des cycles pour la validation finale.

Les cycles quand a eu, sont dés succession de Booking d’endroit et de négociation pour aboutir à un évènement planifier si non on recommence jusqu’à ce qu’on aboutisse à un événement planifié

D’un autre côté, tout ce cursus est décomposé en tâche ; à savoir le Booking contient plusieurs tâches, le cycles de négociation l’organisation ce qui donne suite à une nécessité d’élaborer une autre solution ; celle de management des tâches « TaskManagement ».

Notre besoin s’étend plus, vu que non seulement nous devons prendre en considération l’accessibilité de cette application à travers d’autres mais aussi la possibilité de l’exploiter ailleurs.

### Objectif du projet

Amaris a détecté une nécessité et un besoin fort pour sa bonne gestion de ses ressources matériels et physique et pour la préservation d’une image notable à l’échelle mondiale.

L’organisation d’événements où nous allons nous concentrer, devient quasiment impossible pour l’équipe organisatrice.

Pour mener à bien ce projet, il est indispensable de comprendre le fonctionnement d’organisation d’événements : comment, quand, qui inviter ainsi que les types d’événements.

Aujourd’hui Amaris diversifie les types d’événements. Chaque type d’événement a sa spécificité ainsi qu’un cercle d’envergure. En effet il existe des événements où on peut inviter tous les employés mais pour d’autre ça reste dans un cercle bien fermé. Et plus que ça, aujourd’hui pour des types d’événements précis il est nécessaire de prendre en considération des déplacements de certains employés tels que le comité exécutif et les CEO. Pour ce type de déplacement, il est indispensable dans certains cas où ces personnes-là ne se trouvent pas dans le pays concerné par l’évènement de prendre un billet d’avion, taxi, chauffeur, etc. Mais pour d’autres personnes on parle plutôt d’un service de conciergerie (Jet Privée, Chauffeur, Restaurants, …).

Nous pouvons constater alors, que pour un événement quelconque on a plusieurs contraintes, rajoutant à cela qu’un événement est géré par une équipe et que chaque membre de l’équipe a une tâche bien particulière.

Dans ce cas de figure, il devient quasiment impossible pour une personne de bien gérer ces conditions et suivre ces tâches facilement.

Ce projet vise à ce que Amaris ait une vision détaillée et transparente sur toutes ces dépenses, ces faits et gestes que le groupe Amaris a décidé d’alléger le suivie des tâches et rationaliser la gestion d’organisation des événements.

## Solution envisagée

Pour remédier à ces problèmes, et dans le cadre de l’amélioration continue de l’entreprise, nous proposons de concevoir et de développer une application web et une application mobile et combiner les deux applications en une seule.

Une application web et mobile de planning et de suivi des tâches, il s’agit notamment d’un diagramme de Gantt et d’un calendrier avec un Time Line qui permet de :

* Bien visualiser la complexité des ordres de travail affectées.
* Bien organiser l’affectation des tâches.
* Définir un planning réaliste.
* Assurer la communication sur l’avancement des tâches.
* Travailler sur un environnement sécurisé.

Une application pour la gestion d’évènement qui permet :

* Bien organiser la gestion de l’événement.
* Avoir un visuel clair sur les différents cycles de vie de l’événement.
* Assurer la communication sur l’avancement de la demande du client.
* Réduire le temps de planification et la complexité du travail.
* Assurer la communication entre les différentes applications ; le fait de planifier un vol on planifie une réservation d’hôtel, le fait de planifier un événement en outre la position ou en fait la demande implique la réservation d’un billet d’avion et la réservation d’hôtel si nécessaire.

## Choix de la méthodologie

### Etude Comparatif

#### Les processus unifiés

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Les processus unifiés sont le résultat de l’unification, non pas des processus, mais plus exactement Des meilleures pratiques du développement objet.

Il est distingué par les caractéristiques suivantes :

* **Itératif** : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale par des itérations multiples.
* **Piloté par les risques** : les causes majeures d’échec d’un projet logiciel doivent être écartées en priorité.
* **Centré sur l’architecture** : le choix de l’architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du produit est basée sur ce choix.
* **Conduit par les cas d’utilisation** : le processus est orienté par les besoins utilisateurs présentés par des cas d’utilisation.

#### Les méthodes agiles

Les **méthodes agiles** sont des groupes de pratiques de projets de développement en informatique ([conception de logiciel](http://fr.wikipedia.org/wiki/Conception_de_logiciel)), pouvant s'appliquer à divers types de projets. Elles ont pour dénominateur commun l'[*Agile manifesto*](http://fr.wikipedia.org/wiki/Manifeste_agile). Rédigé en 2001, celui-ci consacre le terme d'« *agile* » pour référencer de multiples méthodes existantes. Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles visent la satisfaction réelle du client en priorité aux termes d'un contrat de développement.

Les méthodes agiles reposent sur une structure ([cycle de développement](http://fr.wikipedia.org/wiki/Cycle_de_développement_(logiciel))) commune (**itérative**, **incrémentale** et **adaptative**), **quatre valeurs** communes :

* **L'équipe** : Les individus et leurs interactions, plus que les processus et les outils
* **L'application** : Des logiciels opérationnels, plus qu'une documentation exhaustive.
* **La collaboration** : La collaboration avec les clients, plus que la négociation contractuelle.
* **L'acceptation du changement** : L'adaptation au changement, plus que le suivi d'un plan.

Le but est d’assumer le fait que l’on ne peut pas tout connaître et anticiper quel que soit notre expérience. On découpe alors le projet en itérations plutôt que de tout prévoir et planifier en sachant que des imprévus arriveront en cours de route.

Voici les avantages du développement itératif :

* **Meilleure qualité de la communication :** L’utilisateur à la possibilité de clarifier ses exigences au fur et à mesure
* **Meilleure visibilité :** Le client a eu meilleure visibilité sur l’avancement des travaux
* **Meilleur contrôle de la qualité :** les tests sont effectués en continu.
* **Meilleure détection des risques :** Les risques sont détectés plus tôt.
* **Motivation et confiance de l’équipe :** satisfaction d’atteindre un objectif fixé.
* C**ontrôle des coûts :** le projet peut être arrêté s’il n’y a plus de budget

#### Tableau comparatif

Voici un tableau récapitulatif des différences entre les méthodes traditionnelles et les méthodes AGILE :

|  |  |  |
| --- | --- | --- |
| Thème | Approche traditionnelle | Approche agile |
| Cycle de vie | En cascade ou en V, sans rétroaction possible, phase séquentielle. | Itératif et incrémental. |
| Planification | Prédictives, caractérisée par des plans plus ou moins détaillées sur la base d'un périmètre et d'exigences définies et stables au début du projet. | Adaptative avec plusieurs niveaux de planification (macro- et micro planification) avec ajustement si nécessaires au fil de l'eau en fonction des changements survenues. |
| Documentation | Produite en qualité importante comme support de communication, de validation et de contractualisation. | Réduites aux stricts nécessaires au profit d'incréments fonctionnels opérationnels pour obtenir le feedback du client. |
| Equipe | Une équipe avec des ressources spécialisées, dirigée par un chef de projet. | Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef de projet. |
| Qualité | Contrôle de qualité à la fin du cycle de développement. | Un contrôle qualité précoce et permanente, au niveau du produit et du processus. |
| Changement | Résistance voire opposition au changement. | Accueil favorable au changement inéluctable, intégré dans le processus. |
| Suivi de l'avancement | Mesure de la conformité aux plans initiaux. | Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire. |
| Gestion des risques | Processus distinct rigoureux de gestion des risques. | Gestion des risques intégrés dans le processus global, avec responsabilisation de chacun dans l'identification et la résolution des risques. |
| Mesure du succès | Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité. | Satisfaction client par la livraison de valeur ajoutée. |

Tableau 1: Tableau comparative entre les méthodes classiques et les méthodes agiles

#### Synthèse

Les méthodes agiles seront plus utilisées pour les gros projets car elles offrent une meilleure adaptabilité, visibilité et gestion des risques. Elles pourraient tout aussi bien être utilisées pour les projets où il n’y pas de documentations détaillées, le client peut alors voir l’évolution du projet et l’adapter selon ses besoins.

En revanche, les méthodes classiques seront plus utilisées s'il y a une idée très précise du projet avec un cahier des charges et planning très détaillé où tous les risques possibles sont anticipés.

### SCRUM

SCRUM est issu des travaux de deux des signataires du Manifeste Agile, Ken Schwaber et Jeff Sutherland, au début des années 1990.Il appartient à la famille des méthodologies itératives et incrémentales et repose sur les principes et les valeurs agiles. Le plus souvent, les experts de SCRUM, même ses fondateurs, le décrivent comme un cadre ou un patron de processus orienté gestion de projet et qui peut incorporer différentes méthodes ou pratiques d’ingénierie. S’il est difficile de définir la nature de SCRUM, sa mise en place est beaucoup plus simple et peut être résumée par la

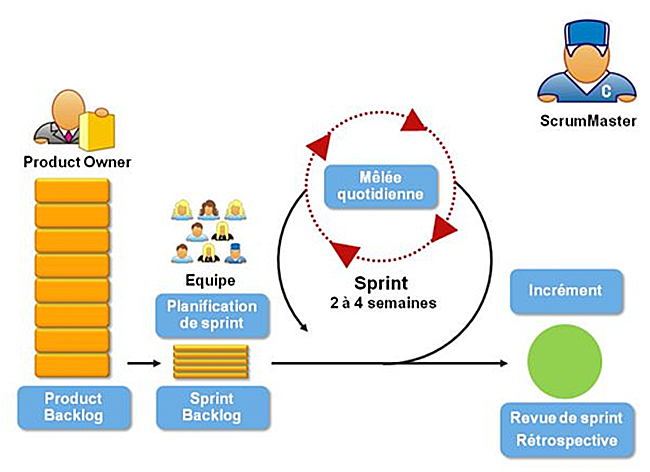


Figure 6 Processus SCRUM

### Pourquoi SCRUM ?

Dans notre cas pour les deux applications ; à savoir « TaskManagementAPI » et « Lifestyle » nous n’avons pas de pas une documentation détaillée et le client n'a pas encore une vue claire des fonctionnalités, le développement doit être donc en collaboration avec le client.

On peut donc justifiez notre choix de SCRUM comme une méthodologie de pilotage pour notre projet comme suit :

* Plus d'interaction avec le client : le client a la possibilité d'adapter, pendant le développement, le logiciel aux besoins métier.
* La grande capacité d’adaptation au changement grâce à des itérations courtes. (TaskManagementAPI a vu le jours grâce à ce changement et n’aurait jamais eu la chance de voir le jour sans l’avantage de SCRUM)
* Equipe réduite de projet : une bonne équipe SCRUM est une équipe assez réduite.

Conclusion

Tous au long de ce chapitre, nous avons mis en évidence les différentes problématiques relatives à l’atteinte de l’information ainsi que les solutions proposées, ce qui a donné naissance à notre projet.

# Chapitre 3 Analyse et spécification des besoins

## Introduction

L’analyse des besoins est une étape primordiale dans le cycle de vie de tout processus et spécialement celui d’un logiciel. Elle permet, de dégager les fonctionnalités attendues de notre future plateforme.

Ainsi nous présentons dans ce chapitre une phase de capture des besoins, suivie par une étape de conception dans laquelle nous établirons l’architecture système et plateforme de développement. Mais avant de venir à ce stade, nous commencerons ce chapitre par l’étude du cadre des besoins.

La définition des besoins permettra de cadrer notre projet assurer sa réussite. A l’issue de cette étape, nous identifierons les différents besoins de chacun des utilisateurs.

## Identification des acteurs

Voici les différents acteurs qui vont interagir avec nos applications :

* **Requestor :** Cet acteur a pour rôle de faire la demande d’un évènement pour un client (interne ou externe).
* **Allocator :** A l’issue de la demande d’événement par le Requestor, l’Allocator vérifie les informations relatives à l’évènement et choisie de valider en assignant un Owner ou bien de rejeter la demande du Requestor pour faute de budget ou bien de marketing
* **Owner :** Une fois assigné, ce dernier commence par affecter des membres pour la gestion de l’évènement, il commence par choisir un Booking agent, un Negociator et éventuellement des membre « TeamMember », il assigne des tâches pour chacun d’eux et peut faire le suivie de l’avancement des tâches**.**
* **BookingAgent :** Ce dernier commence le cycle de recherche et de Booking pour voir les éventuels possibilité et endroit pourl’événement. Une fois il a trouvé il envoie ses résultats au Negociator.
* **Negociator :** le Negociator entame un cycle de négociation avec les gérants des lieux ou des agences pour aboutir à une liste de proposition qu’on soumet au Requestor pour validation, si ce dernier décline les offres en recommence tout le processus.
* **TeamMember :** Tous les acteurs sont par défaut des TeamMember, sauf que pour ce rôle-là, il ne privilégie pas des droits des autres acteurs. Mais cet acteur bénéficie de quelques droits tels que la consultation des événements à venir ou bien la possibilité de contribuer à la réalisation d’un événement par leur contribution physique tout en spécifiant leurs tâches.

## Identification des besoins

Au niveau de cette rubrique nous allons définir les différents besoins fonctionnels et non fonctionnels de notre application.

### Besoins fonctionnels

Vu qu’on a choisie SCRUM comme méthodologie et vu qu’on n’a pas de spécification finale pour ces projets, nous n’avons pas pu identifier tous les besoins fonctionnels, c’est pour cette raison que nous allons présenter uniquement les besoins détecter à ce jour ;

* **Création d’une demande d’évènement**
* **Validation de la demande**
* **Vérification de la disponibilité de l’Owner**
* **Assignation d’Owner**
* **Vérification de la disponibilité des différents membres de l’équipe organisatrice**
* **Assignation des membres de l’équipe organisatrice (BookingAgent, Négociator, Team Member)**
* **Gestion des tâches**

### *Besoins non fonctionnels*

Bien qu’ils ne soient pas en relation avec le métier, les besoins non fonctionnels sont tout aussi essentiels et assurent une meilleure qualité de la solution.

Notre application devra assurer :

* **Sécurité** : L’outil doit être sécurisé et contrôlé par les droits d’accès des utilisateurs. L’utilisation de Clips et des ACL est un moyen crée et utilisée par Amaris.
* **Fiabilité** : Les services offerts doivent fournir des résultats corrects.
* **Maintenabilité** : La maintenabilité et l’évolutivité sont des priorités. Le code sera lisible, commenté, divisé en fonction des pages (des interfaces) et en fonction des tâches abordées.
* **Disponibilité** : L’application sera accessible à travers d’autre application c’est ce qui crée la différence de l’ERP d’Amaris des autres ERP.

## Diagramme de cas d’utilisation

Le diagramme de cas d’utilisation présent définis les fonctionnalités dont dispose l’utilisateur par notre application.

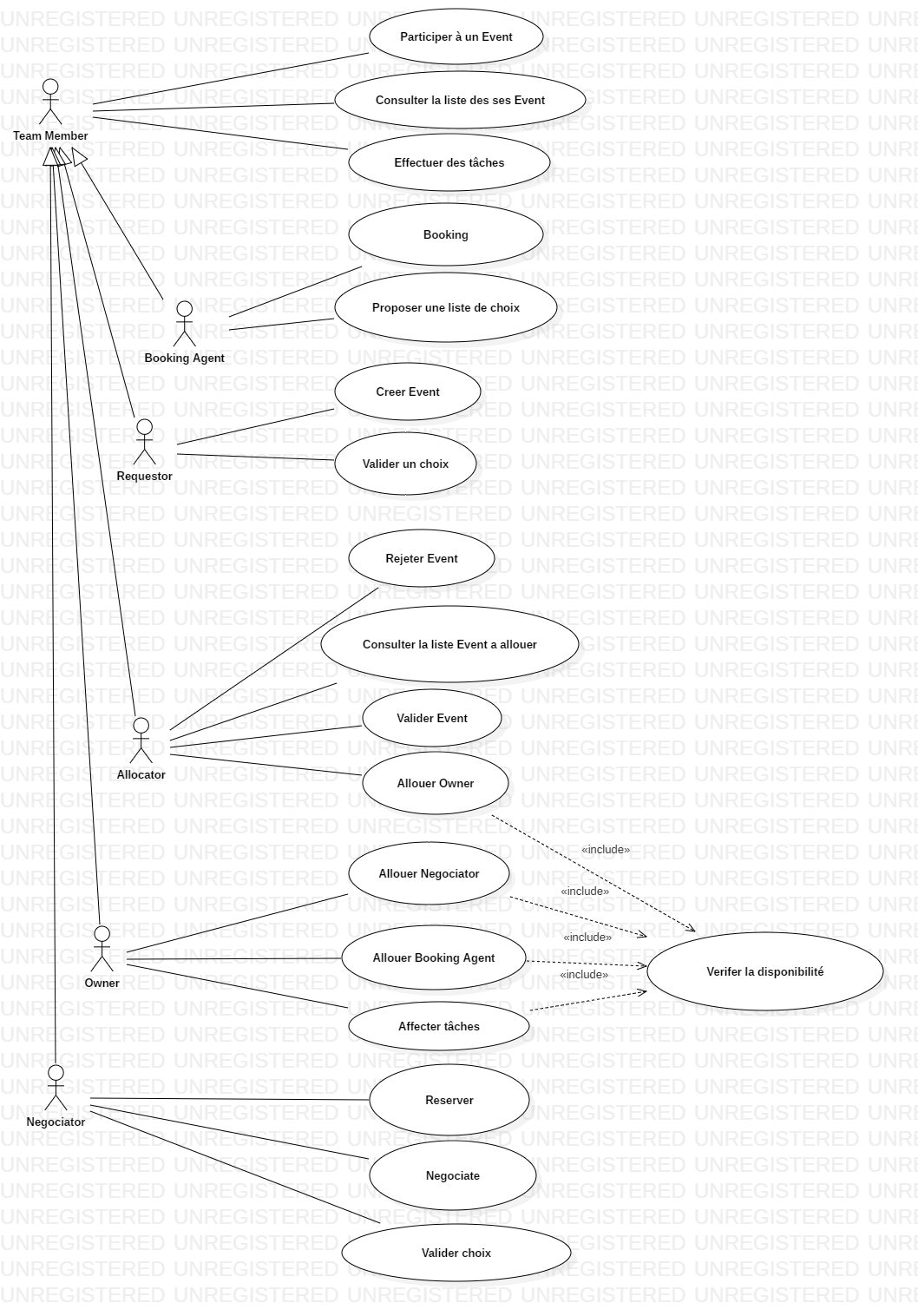


Figure 7 Diagramme de cas d'utilisation général

## Scénario des cas d’utilisation

|  |  |
| --- | --- |
| **Nom** | **Demande de création d’un événement** |
| **Acteur** | **Requestor** |
| **Précondition** | **Utilisateur connecter et éligible à faire une demande pour un évènement** |
| **Post condition** | **La demande a été enregistrés en base de données.**  **Le Requestor a été notifie**  **L’Allocator a été notifier.** |
| **Scénario nominal** | 1. **L’utilisateur clique sur le buttons de création d’événement.** 2. **L’utilisateur rempli le formulaire.** 3. **Le Requestor choisie le client parmi une liste de choix.** 4. **Le Requestor Valide son Formulaire** 5. **Il attend la confirmation de l’événement** 6. **Il reçoit une liste de suggestion d’endroit.** 7. **Il choisit parmi cette liste.** |
| **Scenario alternatif** | **Evènement non Valider**   1. **Il renouvelle une demande d’un autre événement si souhaité** |
| **Exception** |  |

Tableau 2 Description détaillé sur la demande d'évènement.

|  |  |
| --- | --- |
| **Nom** | **Validation de l’évènement et allocation d’un Owner** |
| **Acteur** | **Allocator** |
| **Précondition** | **Utilisateur connecter et a le rôle Allocator et a des évènement a alloué** |
| **Post condition** | **La demande a été enregistrés en base de données.**  **L’Owner a été notifie**  **L’état de l’événement a été changer.** |
| **Scénario nominal** | 1. **L’Allocator consulte les événements non alloués.** 2. **L’Allocator consulte les détails relatifs à l’événement.** 3. **Il assigne un Owner** |
| **Scenario alternatif** | 1. **Evènement non valide il rejette en remplissant le motif.** |
| **Exception** |  |

Tableau 3 Description détaillé sur l'allocation d'un Owner.

|  |  |
| --- | --- |
| **Nom** | **Assignation des membres de l’équipe qui vont organiser l’événement** |
| **Acteur** | **Owner** |
| **Précondition** | **Utilisateur connecter et est un Owner** |
| **Post condition** | **La demande a été enregistrés en base de données.**  **Membres ont été notifie**  **L’état de l’événement a été changer.** |
| **Scénario nominal** | 1. **L’Owner vérifie la disponibilité du BookingAgent.** 2. **L’Owner Affecte un BookingAgent** 3. **L’Owner vérifie la disponibilité du Négociator** 4. **L’Owner affecte un Négociator** 5. **L’Owner vérifie si l’événement nécessite d’autre membre.**    1. **Il vérifie la disponibilité des membres**    2. **Assigne des Team Membre.** 6. **L’Owner affecte des tâches à chaque membre.** |
| **Exception** |  |

Tableau 4 Description détaillé sur l'assignation de l'équipe évent.

|  |  |
| --- | --- |
| **Nom** | **Faire la recherche pour les différentes possibilités pour la réalisation de l’événement** |
| **Acteur** | **BookingAgent** |
| **Précondition** | **Utilisateur connecter et est un BookingAgent et est assigné comme BookingAgent a un évent** |
| **Post condition** | **Liste de choix valide a envoyé pour le Requestor** |
| **Scénario nominal** | 1. **Le BookingAgent consulte son carnet.** 2. **Vérifie la disponibilité de l’endroit.** 3. **Donne multiple choix.** 4. **Envoie cette liste au Negociator pour valider.** |
| **Scenario alternatif** | **La liste de choix n’est pas validée :**  **Refaire tout le processus à nouveau jusqu’à aboutir à un choix validé** |
| **Exception** |  |

Tableau 5 Description détaillé sur la recherche de proposition

|  |  |
| --- | --- |
| **Nom** | **Négociation avec les commerces** |
| **Acteur** | **Negociator** |
| **Précondition** | **Utilisateur connecter et est un Negociator et est assigné comme Negociator a un évent** |
| **Post condition** | **Réservation suite à la négociation.** |
| **Scénario nominal** | 1. **Consulte les différents choix du Requestor.** 2. **Entame des cycles de négociation avec les responsables des lieux.** 3. **Valide avec l’équipe finance pour allouer le budget.** 4. **Reserve l’endroit.** 5. **Notifie le Requestor.** |
| **Scenario alternatif** | **Refaire un autre cycle de négociation.** |
| **Exception** |  |

Tableau 6 Description détaillé sur la négociation.

|  |  |
| --- | --- |
| **Nom** | **Effectuer Tâche** |
| **Acteur** | **Team Member** |
| **Précondition** | **Assignation comme membre** |
| **Scénario nominal** | 1. **Consulter les différents acteurs** 2. **Consulter la liste des événements** 3. **Consulter la mission** 4. **Découper la mission en tâches** 5. **Mettre à jours l’état d’avancement de chaque tâche** |
| **Exception** |  |

Tableau 7 Description détaillé sur la Team Member.

## Diagramme de classe Analyse

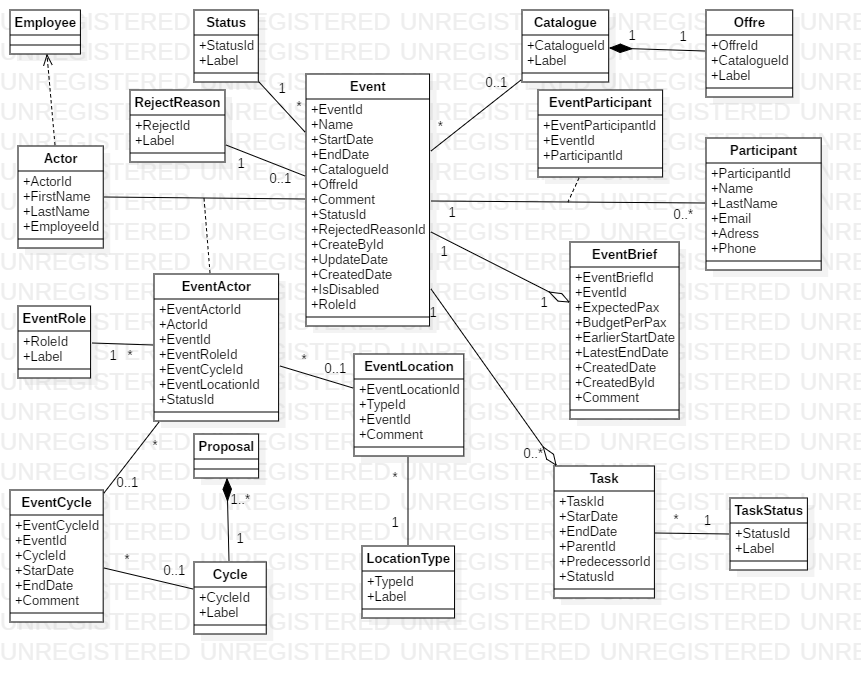


Figure 8 Diagramme de classe d'analyse

## Diagramme de séquence Système :

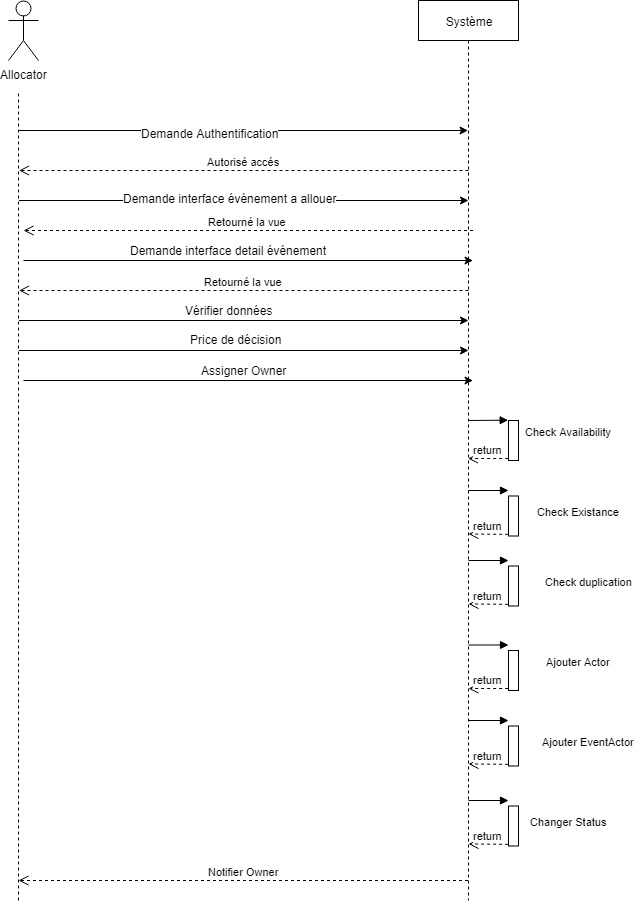


Figure 9 Diagramme séquence Systéme

## Maquettes

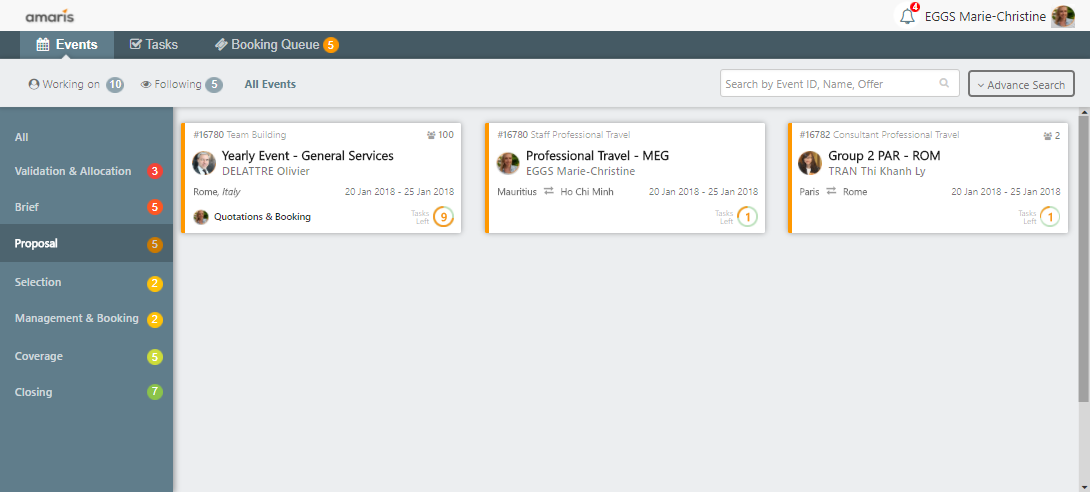


Figure 10 Maquette liste des Proposal

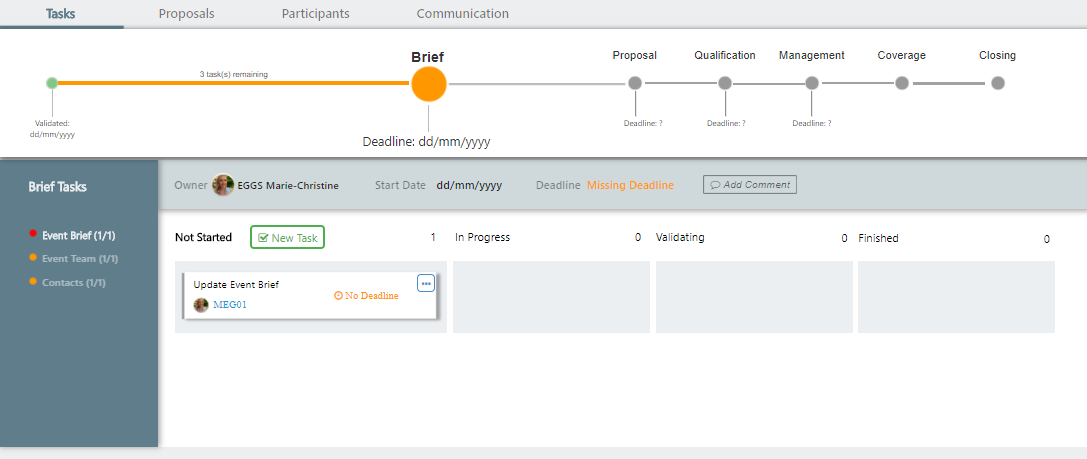
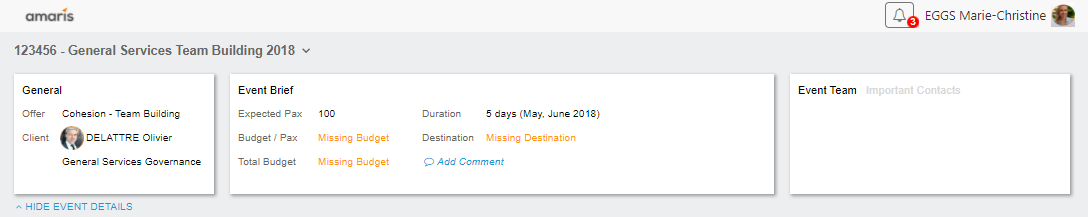


Figure 11 Maquette Event Brief

## Backlog du produit

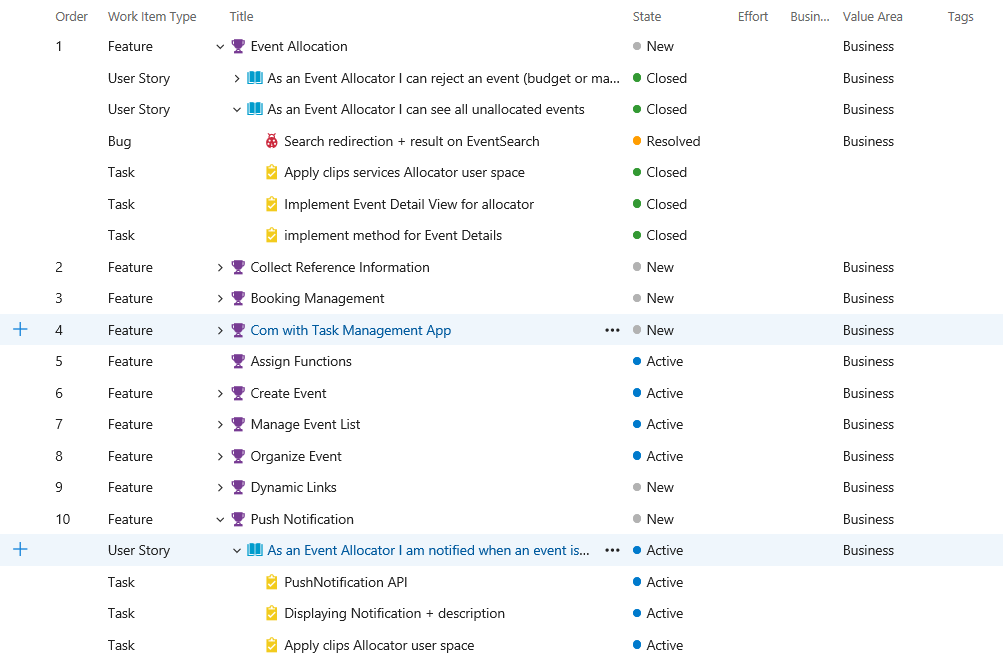


Figure 12 Backlog du produit LifeStyle sur TFS

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Type** | **State** | |
| Event Allocation | Feature | | New |
| As an Event Allocator I can reject an event (budget or marketing) | User Story | | Closed |
| As an Event Allocator I can see all unallocated events | User Story | | Closed |
| Com with Task Management App | Feature | | New |
| Assign Functions | Feature | | Active |
| Create Event | Feature | | Active |
| As an authorized user I can request an event | User Story | | Closed |
| Manage Event List | Feature | | Active |
| As a Director/Manager/Senior I can see all events by owner in my scope | User Story | | New |
| As a Director/Manager/Senior I can check details of an event in my scope | User Story | | New |
| As a Director/Manager/Senior I can modify details of an event on behalf of the owner in my scope | User Story | | New |
| Organize Event | Feature | | Active |
| As an Event Allocator I can allocate an Event Owner on an Event | User Story | | Closed |
| As an Event Owner I can manage the participant (employees only) list of my events | User Story | | Closed |
| As an Event Owner I can add team members on my events | User Story | | Closed |
| As an Event Owner I can assign multiple members to a role in the Event Team | User Story | | Closed |
| As an Event Owner I can update all my Events details | User Story | | Closed |
| As a member of the Event Team I can see the event details | User Story | | New |
| Push Notification | Feature | | New |
| As an Event Allocator I am notified when an event is created without an owner | User Story | | Active |
| Manage Proposal Cycle | Feature | | New |
| As an Event Requestor I can modify my Open Request | User Story | | New |
| Data Base Refactoring | Feature | | Active |
| Code Refactoring | Feature | | Active |

Tableau 8 Exemple Backlog (Lifestyle)

## Architecture Physique

Notre application va reposer sur une architecture 3-tiers. Nous aurons un serveur pour la Base de données, un serveur Web pour traiter les requêtes du client et un serveur client.

Nous avons opté pour cette solution pour plusieurs raisons à savoir :

La base de données est déployée sur un serveur distant vu que celle de notre ERP n’est pas séparée pour l’ensemble de ses applications, en effet tous les schémas ainsi que les tables sont implémentés au niveau d’une seule base de données, tandis que le code source et éparpiller un peu partout dans le monde.

Mais aussi, nous justifiant ce choix par le biais de la haute disponibilité, en effet le fait d’avoir un serveur de backup et le stockage sur le cloud nous permet de l’assurer.

Nous avons aussi d’autre serveur IIS pour assurer le Load balancing. La figure suivante illustre notre architecture :

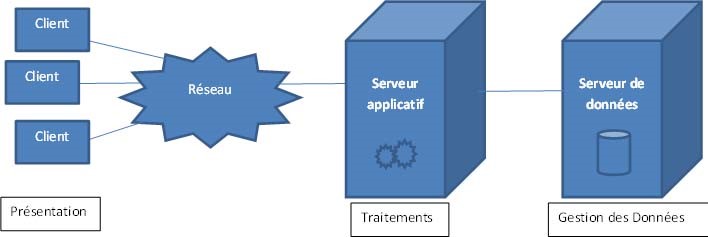


Figure 13 Architecture Physique

## Architecture Logique

L’architecture de notre projet se compose de 5 projets différents, chacun apporte sa touche personnelle pour assurer le bon fonctionnement du projet.

* **LifeStyle** : La solution web du projet qui se compose de vues et de contrôleurs.
* **LifeStyle.API.Client** : Cette solution permet aux clients de consommer les données à partir d’une solution web API existante.
* **LifeStyle.Service** : Tous les services se trouvent dans cette solution.
* **LifeStyle.Contract** : Les interfaces seront créées dans cette solution, ces derniers nous permette d’écrire le contrat de chaque méthode.
* **LifeStyle.Model** : Les Models seront créés dans cette solution, les modèles nous permettent d’alléger la quantité d’information qu’on veut avoir à partir de la DAL.
* **LifeStyle. DAL** : Toutes les classes qui ont un lien direct avec la base de données seront créées dans cette solution.

La figure ci-dessous nous permet de mieux comprendre notre architecture logique est les différentes couches de celle-ci.

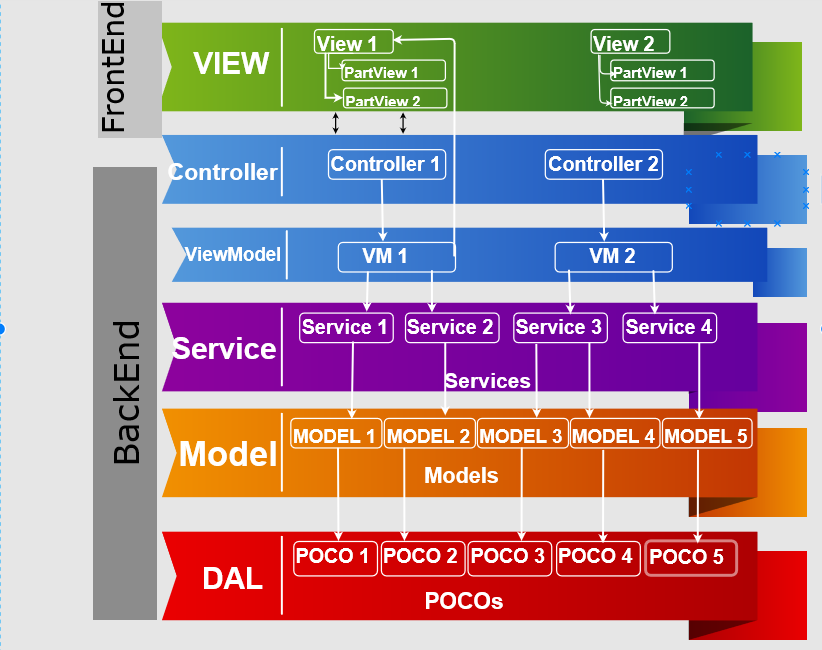


Figure 14 Architecture logique

## Base de données

Le choix de la base de données à adopter nécessite une étude approfondie et la compréhension des produits et des applications du ERP du groupe.

Suite à cette étude, nous avons privilégié le choix de la base de données SQL vu qu’elle s’est avérée plus appropriée pour notre application du fait de son temps de réponse rapide et de sa robustesse face aux données volumineuses.

Une deuxième étude nous a permis de valider ce choix, vu que notre application sera accessible par d’autre applications. En effet l’ERP d’Amaris a un avantage celui de la communication entre les différents supports et applications ce qui fait une nécessité que toutes les données soient structurées sur une même base de données, certes l’utilisation des API simplifie et remédie à ce problème sauf que certaines informations et certaines anciennes applications ne nous fournissent pas d’API ce qui nous laisse dans l’obligation de nous orienter vers ce choix.

## Diagramme de déploiement

C’est le premier niveau de conception car il permet d’organiser les environnements de travail sur le réseau. Pour cela nous modélisons notre architecture par un diagramme de déploiement décrivant la répartition des différents composants et leurs interactions en termes de communication.



Figure 15 Diagramme de déploiement

Il propose une vision statique de la topologie du matériel sur lequel s’exécute le système, il montre les associations (connexions) existantes entre les nœuds du système et les interactions entre eux.

Dans le cas de notre application, nous avons dégagé quatre nœuds principaux :

**Couche Présentation** : c’est la couche qui correspond à l’affichage, la restitution sur le poste de travail et le dialogue avec l’utilisateur.

**Couche Web** : c’est la couche qui assure la communication entre le client et l’application et qui de ce fait contient un serveur à travers des requêtes http, le serveur web utilisé est le IIS.

**Couche application** : Contient l’application ASP.NET responsable de l’accès aux données, des calculs sur les données et de leur validation

**Couche données** : Cette couche contient la base de données.

Conclusion

La capture des besoins fonctionnels, non fonctionnels, les maquettes, l’architecture, le choix de la méthodologie ainsi que l’étude des cas d’utilisations un par un ainsi que l’identification des acteurs est une étape indispensable car elle mène la réalisation du projet dans la bonne voie après avoir fait une bonne conception technique qui va être bien détaillée dans le prochain chapitre.

# Chapitre 4 Conception

## Introduction

Dans ce chapitre nous allons nous intéresser aux aspects de la conception tout en se basant sur les informations collectées grâce à l’étude de l’existant, la critique ainsi que la problématique

Le but d’établir les diagrammes nécessaires au développement de l’application, est de pouvoir définir peu à peu les fonctionnalités de bases, ainsi que définir l’architecture choisie pour l’application.

Mais nous devons aussi songer aux patrons de conceptions que nous allons utiliser et définir les points fort de chacun pour qu’ils soient en cohérence et justifie aux besoins ultérieurs d’éventuel utilisation à partir des applications tierces.

## Conception Préliminaire

### Les Patrons de conception

#### Le patron de conception MVC

Conçu pour des applications dites « client lourd », c’est-à-dire dont la majorité des données sont traitées sur le poste client. Ce modèle de conception était tellement puissant qu’il a finalement été adopté comme modèle pour la création d’applications ou de sites web.

Ce design pattern est une solution reconnue permettant de séparer l’affichage des informations (la vue), les actions de l’utilisateur (le contrôleur) et l’accès aux données (le modèle). Il s’agit du principal avantage apporté par ce modèle de conception, qui présente donc une architecture claire et normalisée, qui facilitera également le développement, l’évolutivité et la maintenance des projets. MVC signifie donc Modèle-Vue-Contrôleur

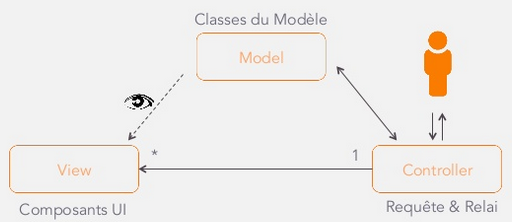


Figure 16 Architecture MVC

#### Le patron de conception Repository

Le repository est fait pour abstraire l'accès aux données. Donc il n'a de sens que dans la récupération de données. Il est basé sur cinq **méthodes** nécessaires à l'accès aux données :

* La sélection d'un enregistrement d'une table via son ID
* La sélection de tous les enregistrements
* La mise à jour
* L'insertion
* La suppression

Cette liste est cependant loin d'être exhaustive, on peut l'étendre à notre convenance afin d'implémenter des méthodes répondant à nos besoins.

## Conception détaillé

### Diagramme d’entité

#### Diagramme de classe de conception

A close up of a newspaper

Description generated with high confidence

Figure 17 Diagramme de classe de conception

## Diagramme de séquence

### Diagramme de séquence Objet (Assign Owner) :

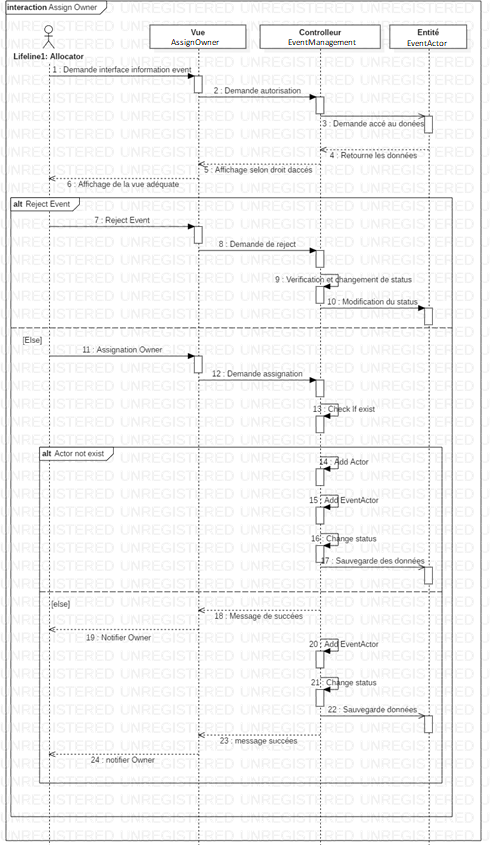


Figure 18 Diagramme séquence objet (Owner)

Lors de son assignation, L’Allocator commence par vérifier l’information relative à l’événement son type, le demandeur de cet événement, est le droit de ce dernier par rapport au type de l’événement.

Par la suite s’il choisit de valider l’événement il assigne un Owner et automatiquement l’événement sera accepté, à ce niveau-là la tâche du Owner commence. Si, dans le cas contraire il refuse l’événement pour l’une des raisons spécifieée préalablement, il doit mentionner la principale cause du rejet.

### Diagramme de séquence Objet (Assign Membre) :

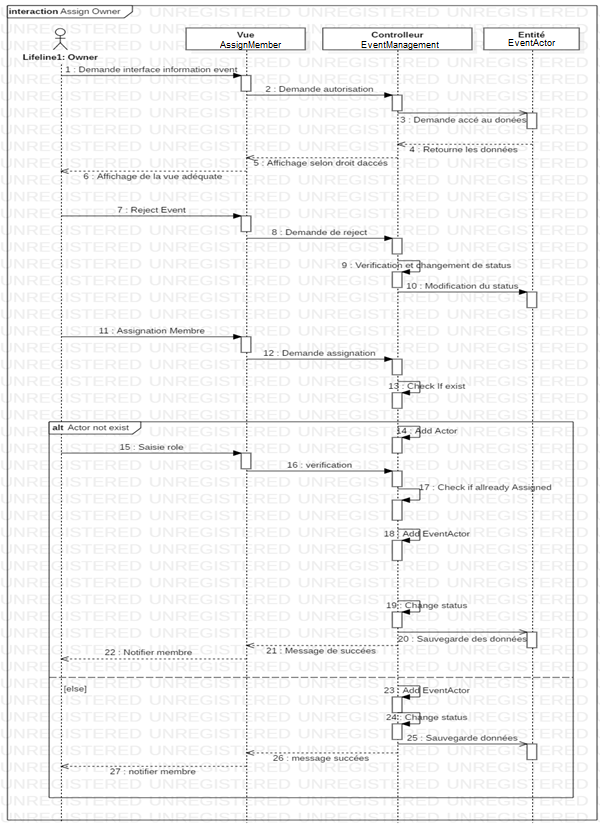


Figure 19 Diagramme séquence objet (Membre)

Dès son allocation comme étant Owner, ce dernier commence par voir la nécessité du Requestor, par la suite il enchaine par spécifier la team adéquate selon la disponibilité de chacun pour certains cas il peut assigner des membres qui ne sont pas disponible vue que ces derniers peuvent déléguer leurs tâches oralement pour leurs collègues.

Il ajoute un ou plusieurs Booking agent et des Negociator. Il peut également rajouter des Team Member pour y contribuer.

### Diagramme de séquence Objet (Request Event) :

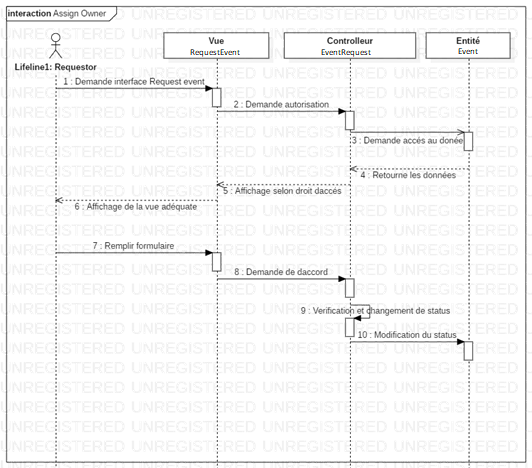


Figure 20 Diagramme séquence objet Requestor

Etant un employé d’Amaris, tout individu a le droit de faire une demande d’un événement, cependant pour des types bien précis selon le cercle d’appartenance et le droit de chacun.

Ce dernier s’authentifie sur l’application et rempli le formulaire adéquat et puis valide.sa demande sera traiter par la suite par l’Allocator. En cas de rejet le Requestor peu à nouveau faire une demande en prenant en considération le motif du refus.

## Diagramme d’état transition

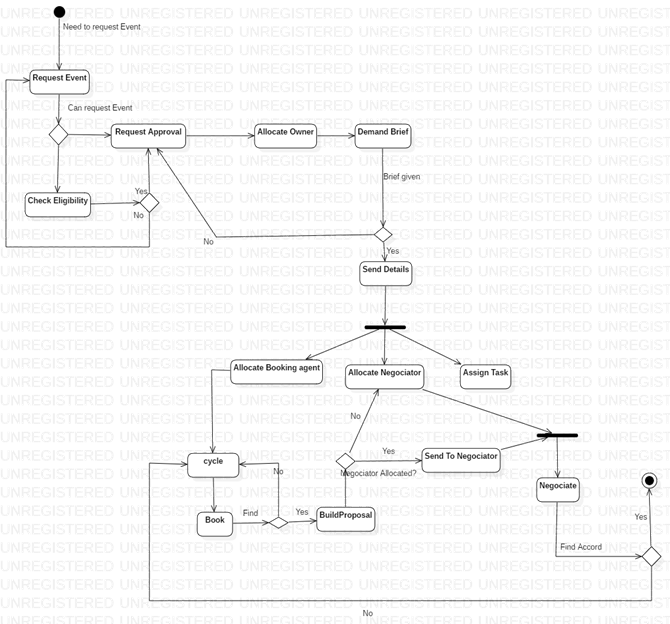


Figure 21 Diagramme d'état transition

Conclusion

La phase de la conception technique présente une étape indispensable et primordiale pour réussir le projet vu que celle-ci va bien mettre en ordre les différentes étapes de la réalisation de ce qui est demandé qui va être justement décrit dans le prochain chapitre avec ses différents détails.

# Chapitre 5 Réalisation

## Introduction

Il ne nous est pas inconnu que, dans un rapport de projet de fin d’études, Il est indispensable de présenter, l’environnement du travail matériel, tout autant, l’environnement du travail logiciel, ainsi que les interfaces finales de l’application Android.

## Environnement du travail

### Environnement logiciel

* SGBD : SQL
* Langage de programmation :HTML,CSS,Javascript,ASP.NET
* Système d’exploitation : Windows 7.

### Langages utilisés

#### HTML

HTML5 « Hyper TextMarkupLanguage » est un langage de balisage web qui a pour objectif l’affichage d’une page web visible à l’égard des internautes. Ce langage nous a justement permis de créer les pages web du site.

#### CSS

C’est un langage qui permet de gérer avec une grande souplesse la présentation des pages web, en effet il présente des feuilles de style « cascading style sheets ».

Les styles servent à définir les règles à appliquer sur un ou plusieurs documents HTML.

Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les [couleurs](http://www.futura-sciences.com/magazines/matiere/infos/dico/d/physique-couleur-4126/), les marges et espacements, les bordures, les images de fond, etc.

CSS vise à séparer la structure d’un document HTML et sa présentation

#### *JavaScript*

JavaScript un langage de script orienté objet principalement utilisé dans les pages. Le développeur Internet code ses pages HTML en y intégrant des sources JavaScript et il peut également créer des objets sur les pages, des propriétés et des actions selon les évènements déclenchés par le visiteur (clic, saisie clavier …) de son côté l’internaute à travers son navigateur charge le code des pages.

#### JQuery

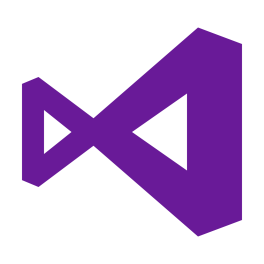
JQuery est une bibliothèque JavaScript qui permet d’ajouter des fonctions supplémentaires comme des animations aux sites web. Il est open source

#### *POSTMAN*

****

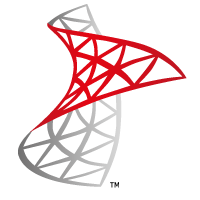
Est une extension du navigateur Google Chrome qui est un outil destiné aux tests des services web et assure les mêmes fonctionnalités que REST Client.

#### VISUAL STUDIO

****Est utilisé essentiellement pour le développement de ses applications en effet la majorité des applications d’Amaris sont en ASP.NET.

.

#### SQL SERVER

****

Amaris a également opté pour ce choix de serveur de base de données pour sa souplesse et la facilité de son utilisation avec le framework ASP.NET

#### ENTITY FRAMEWORK



Fournit un accès à des technologies clés simplifiant le développement  
d'applications web ASP et de services web XML.

.

#### TFS Team Foundations Server

Ces outils sont utilisés par Amaris comme espace de dépôt de planification de suivi et la gestion du développement de ses projets.

#### OUTLOOK

**Est** utilisé comme est un gestionnaire d'informations personnelles et un client de courrier électronique on l’utilise également pour la planification des meetings.

#### SKYPE FOR BUISINES

**** Outils de communications interne est vocale, il nous a été utile pour notre communication directe et les meetings distant ainsi que pour le partage d’écran pour d’éventuel nécessité technique.

## Scénarios

Dans cette partie nous présenterons plusieurs scénarios qui illustrent des exemples de la manière dont passe un événement de sa demande jusqu’à sa réalisation.

Ces scénarios présentent l’ensemble des fonctionnalités de l’application réalisée ainsi que l’interface de chacune par des captures d’écran accompagnent une explication brève de chacune.

### Scénario Request Evénement

Pour pouvoir faire une demande d’un événement, tout d’abord l’utilisateur doit être un employé d’Amaris et qu’il s’identifie. S’il n’est pas autorisé à utiliser cette application la fenêtre suivante s’affiche :

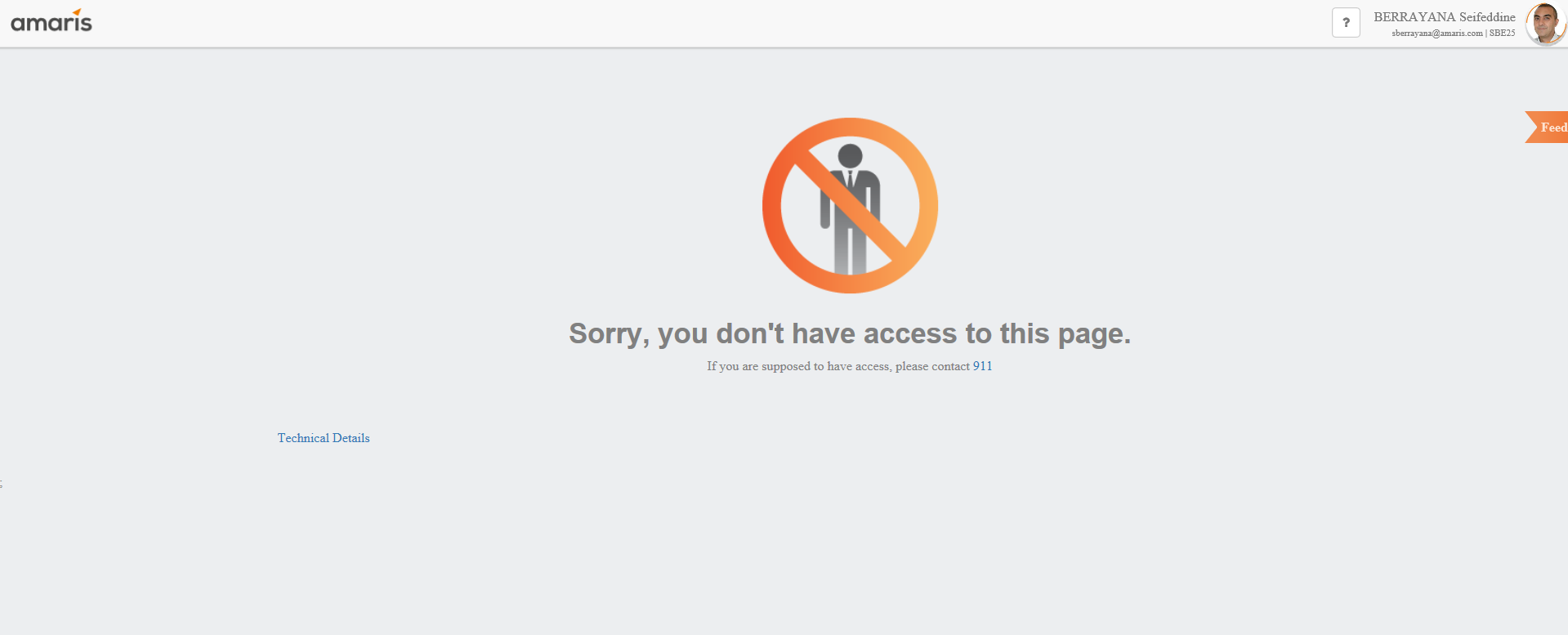


Figure 22 Utilisateur non autorisée a utilisé l'application

Si au contraire il est autorisé a utilisé l’application mais avec certaines restrictions la fenêtre suivante s’affiche, il devra se rendre uniquement dans l’espace où il est autorisé à accéder

N.B : les restrictions se font selon les rôles de chacun.



Figure 23 Utilisateur autorisé à faire une demande uniquement

Une foisqu il il a accédé, ce dernier s’il souhaite créer un événement il se rend à l’interface appropriée (voir figure ci-dessous) où il trouvera sa liste des demandes crée auparavant et il trouvera un bouton Request For Event celui-ci lui affichera une autre interface que nous découvrirons ultérieurement

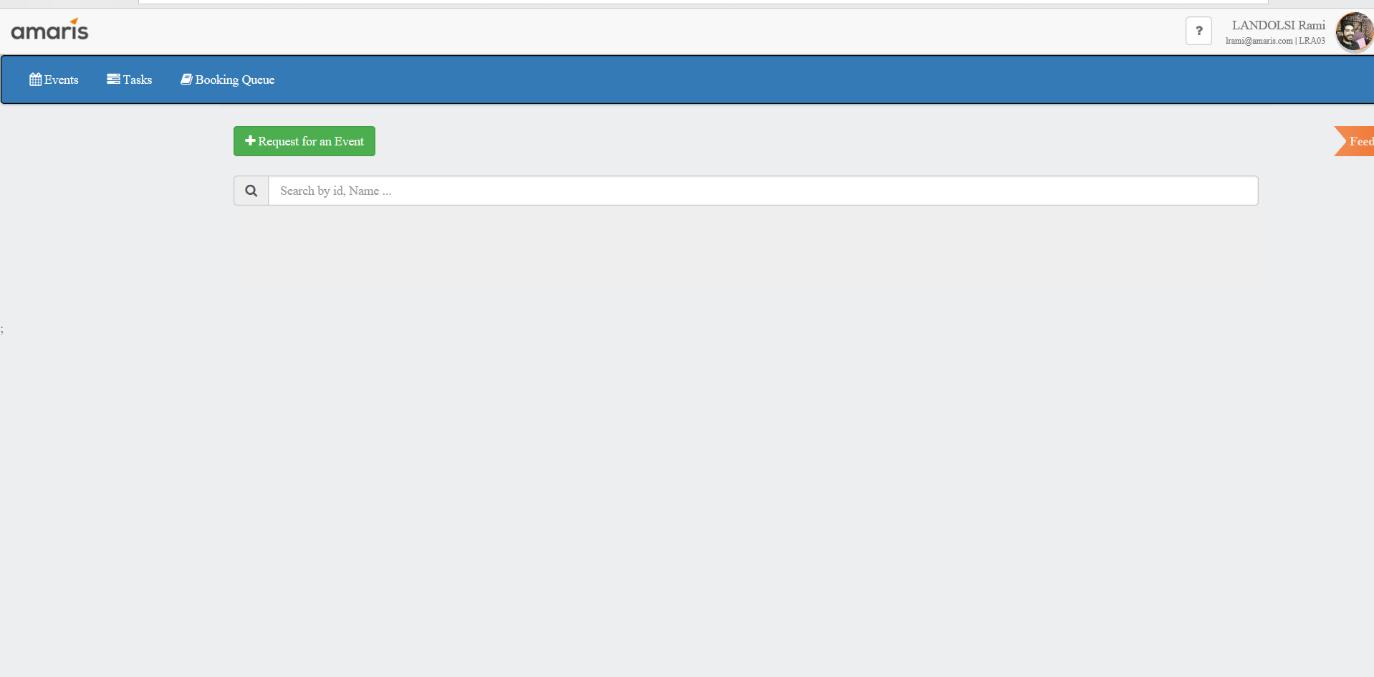


Figure 24 Interface d'accueil

Puis ce dernier commence par se rendre à la rubrique spécifiée, puis il commence par valider son formulaire selon son besoin. Cette figure présente le formulaire approprié

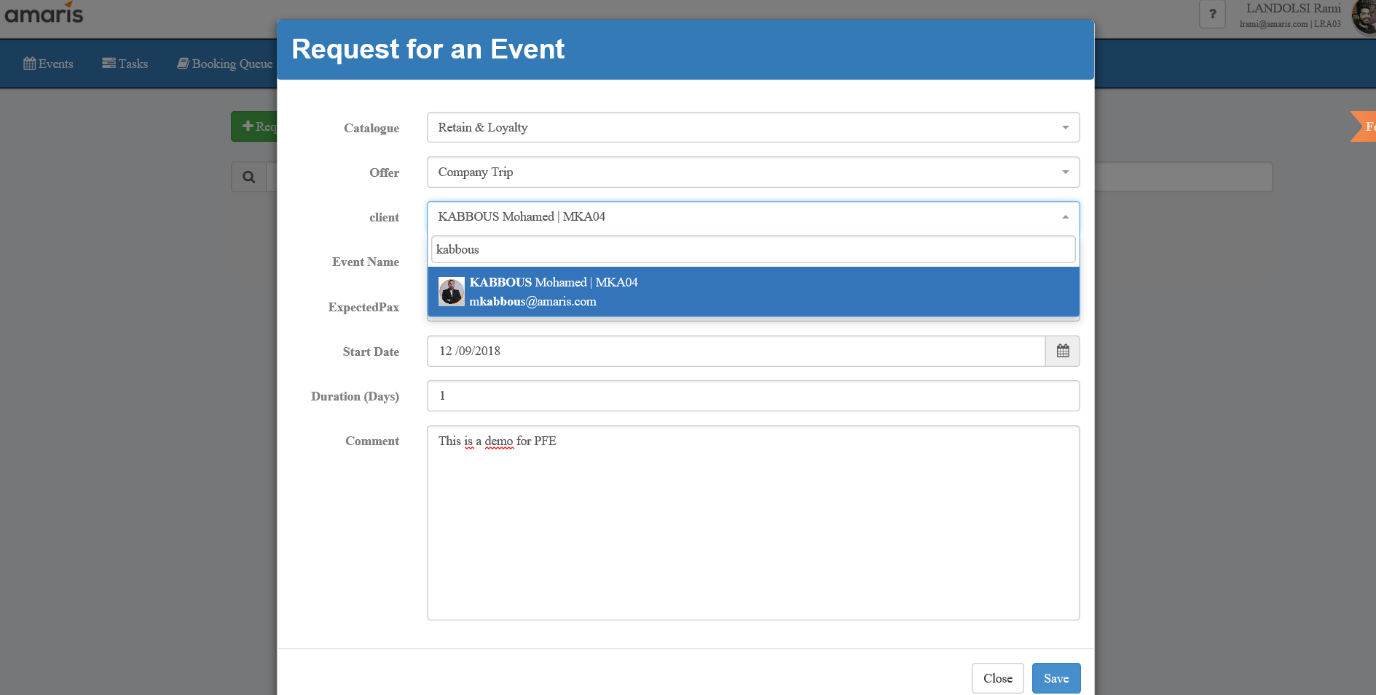


Figure 25 Demande d'un événement

Finalement il finit par valider sa demande qui va être traiter par l’Allocator par la suite.

### Scénario Reject Evénement

Pour le cas du rejet d’un événement l’utilisateur commence par visualiser la liste d’événement à allouer comme l’illustre la figure suivante

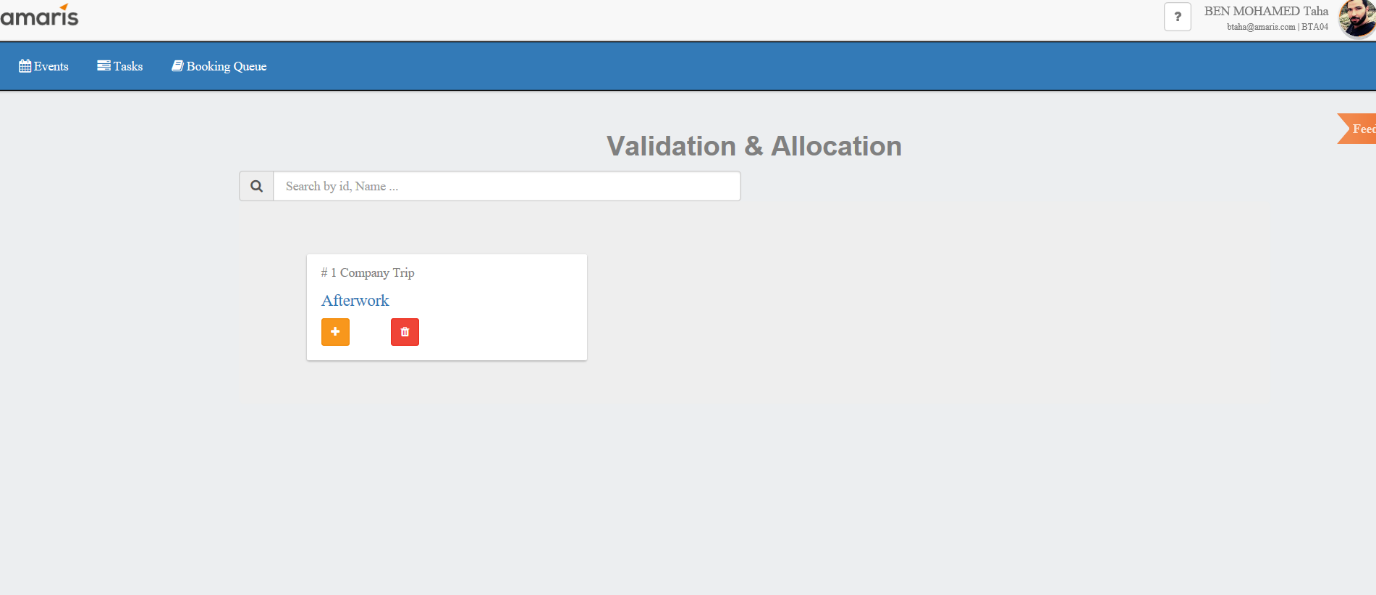


Figure 26 Interface de rejet d'un événement

Il consulte par la suite le détail d’un événement de la liste comme l’illustre cette figure :

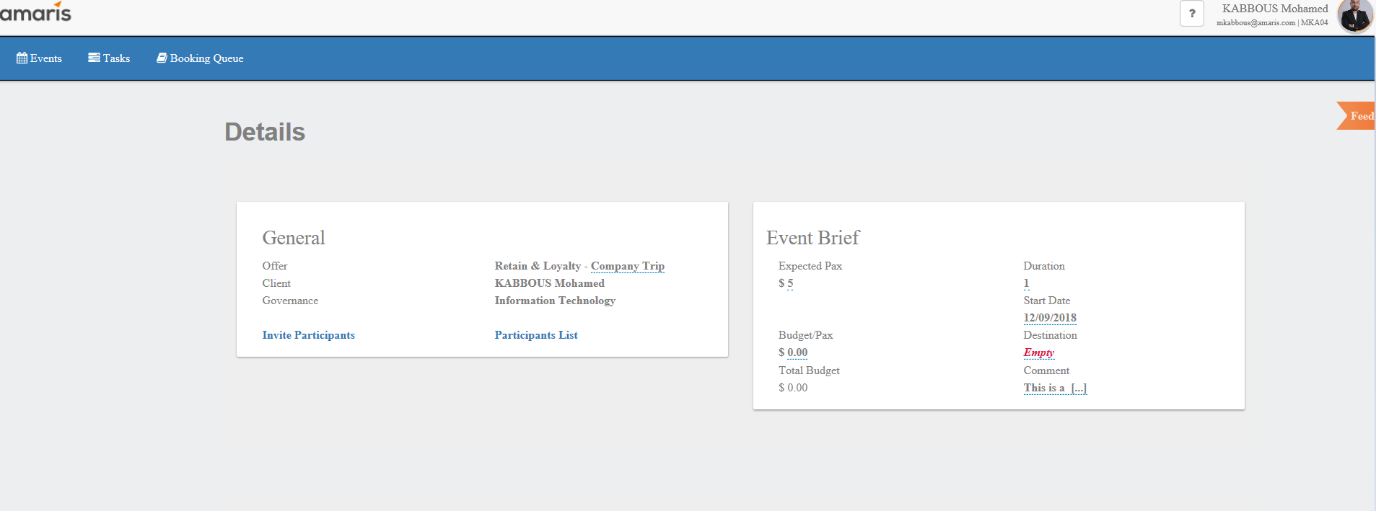


Figure 27 Interface des détails d'un événement

Puis, s’il fait le choix du rejet en remplissant le formulaire ci-dessous en spécifiant le motif du refus avec un commentaire pour le Requestor.

Le motif du refus peut être une des deux possibilités suivantes :

* + 1. Budget : Si le budget est trop élevé pour cette événement ou cette événement n’est pas une obligation pour le groupe et qu’il va couter énormément.
    2. Marketing : Si l’événement n’est pas faisable ou bien l’emplacement choisie relève d’une impossibilité d’ordre commerciale. Ou bien si les négociations avec les partenaires ou bien les commerces n’ont pas aboutie.

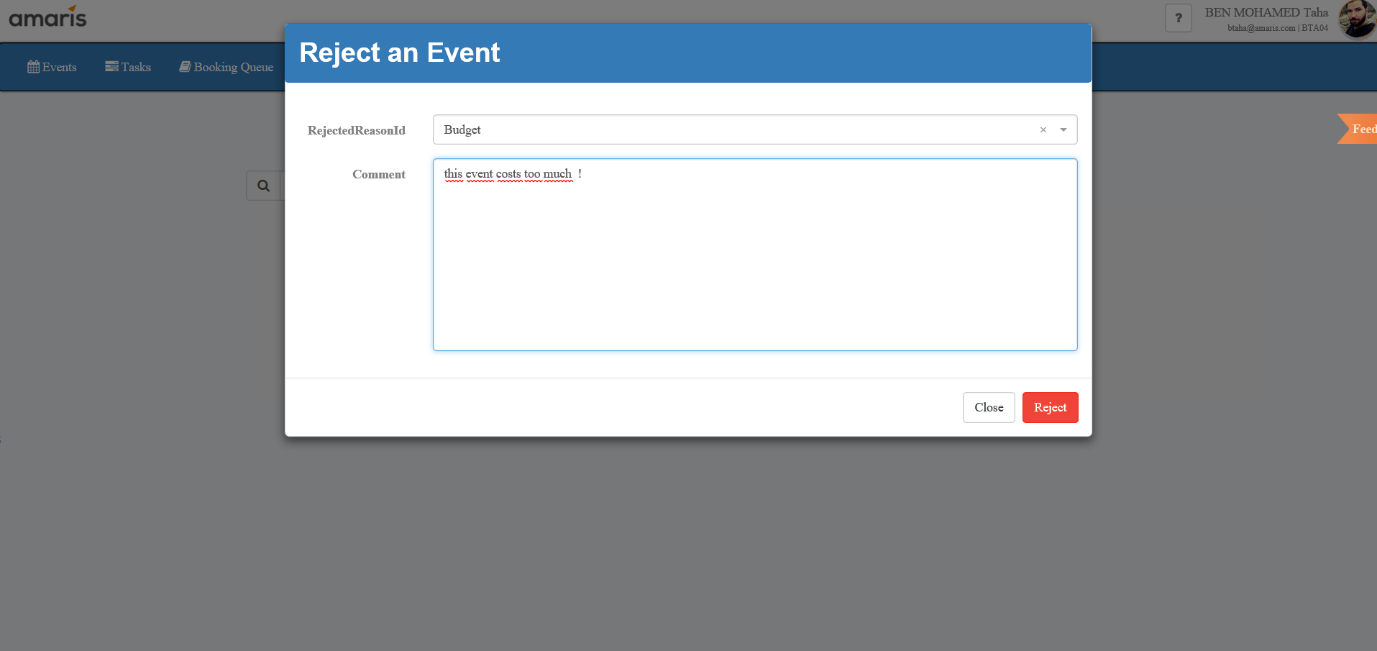


Figure 28 Interface du motifs de rejet d'un événement

### Scénario Allocation Owner pour un événement

Si l’Allocator fait le choix de valider l’événement, il assigne un Owner pour ce dernier, et l’événement sera automatiquement validé.

Il commence par voir la liste des employées il on choisit un comme Owner, puis il vérifie sa disponibilité ; on peut voir sur la figure suivante un espace approprié ou s’affiche la disponibilité des membre intitulé « Member Unavailability »

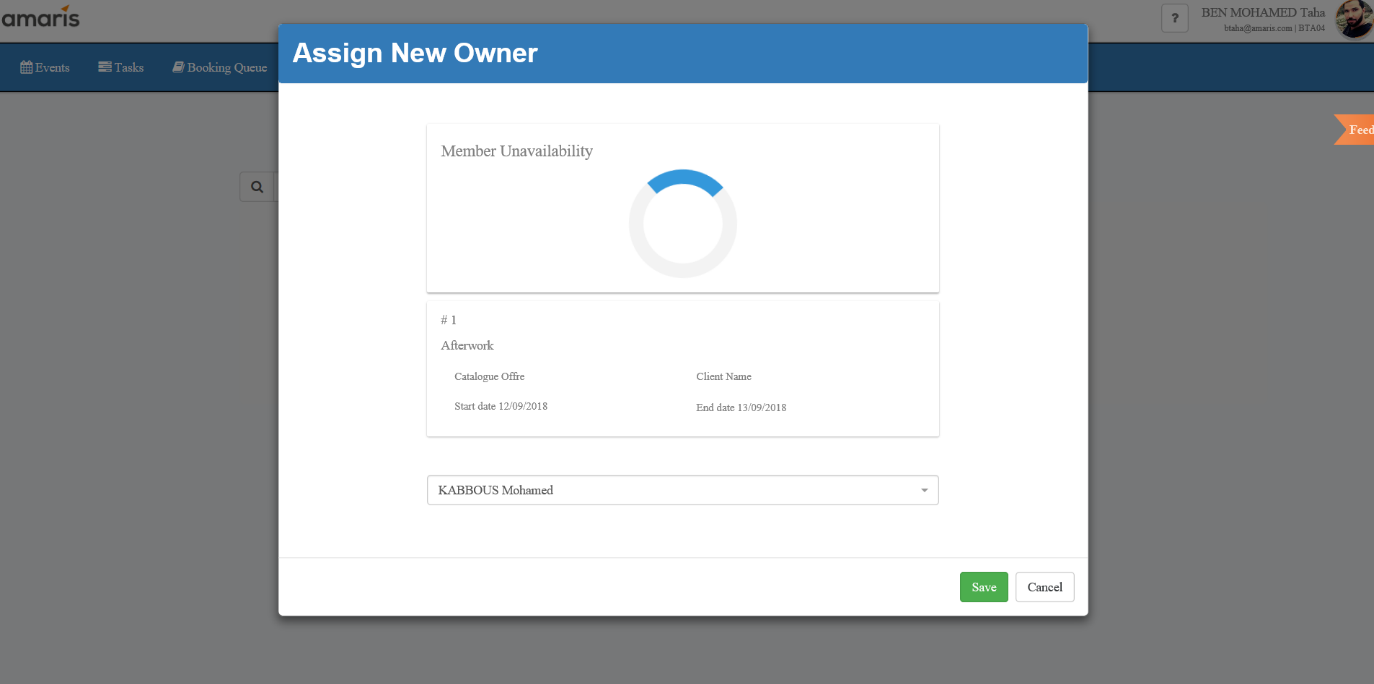


Figure 29 Interface d'assignation d'un Owner

### Scénario Allocation BookingAgent et Négociator pour un Evénement

Les événements sont affichés selon un ordre chronologique (Timeline), les événements les plus urgents sont affichés en premier lieu,

Des informations globales sont affichées sur ce Timeline tel que la photo du Requestor ou bien le nom de l’événement

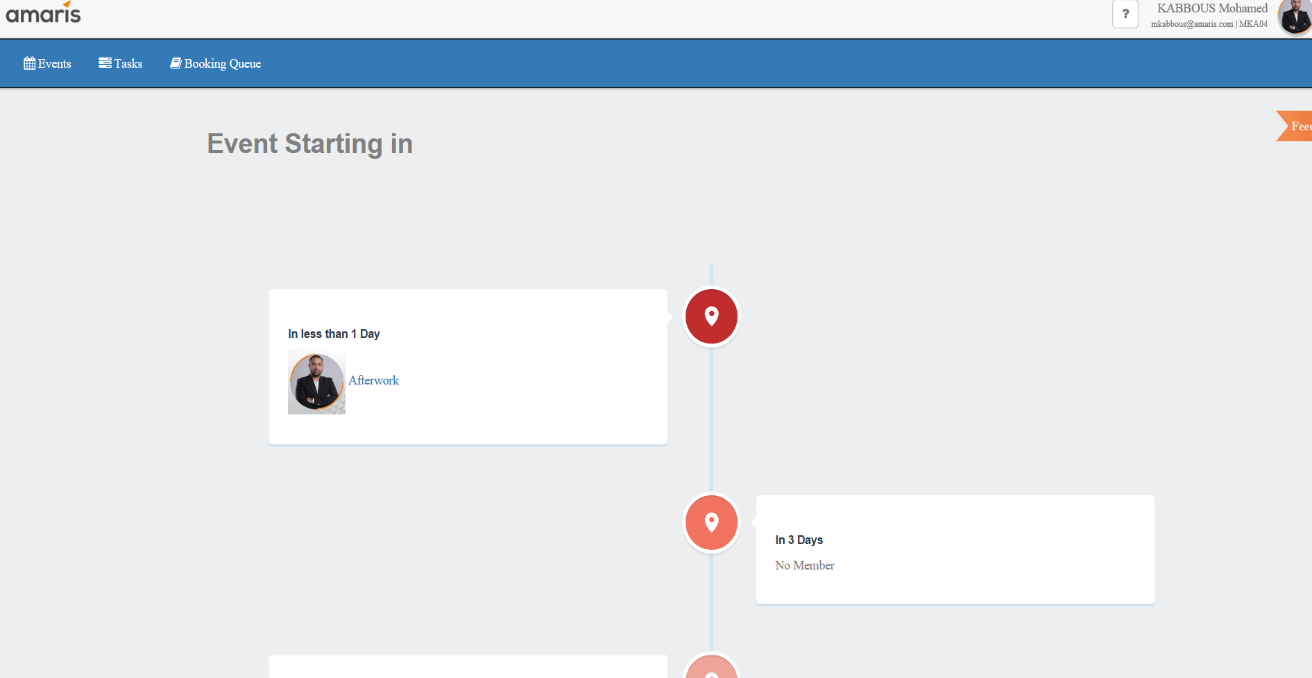


Figure 30 chronologie des événement

Une fois qu’il clique sur le nom d’un événement, une interface s’affiche contenant les détails de ces derniers avec des informations relatives aux participants s’il y’en a ou bien à l’équipe d’organisation Puis il fait le choix du type du membre à ajouter dans notre cas (ce scénario) il fait le choix d’ajouter un BookingAgent ou un Négociator

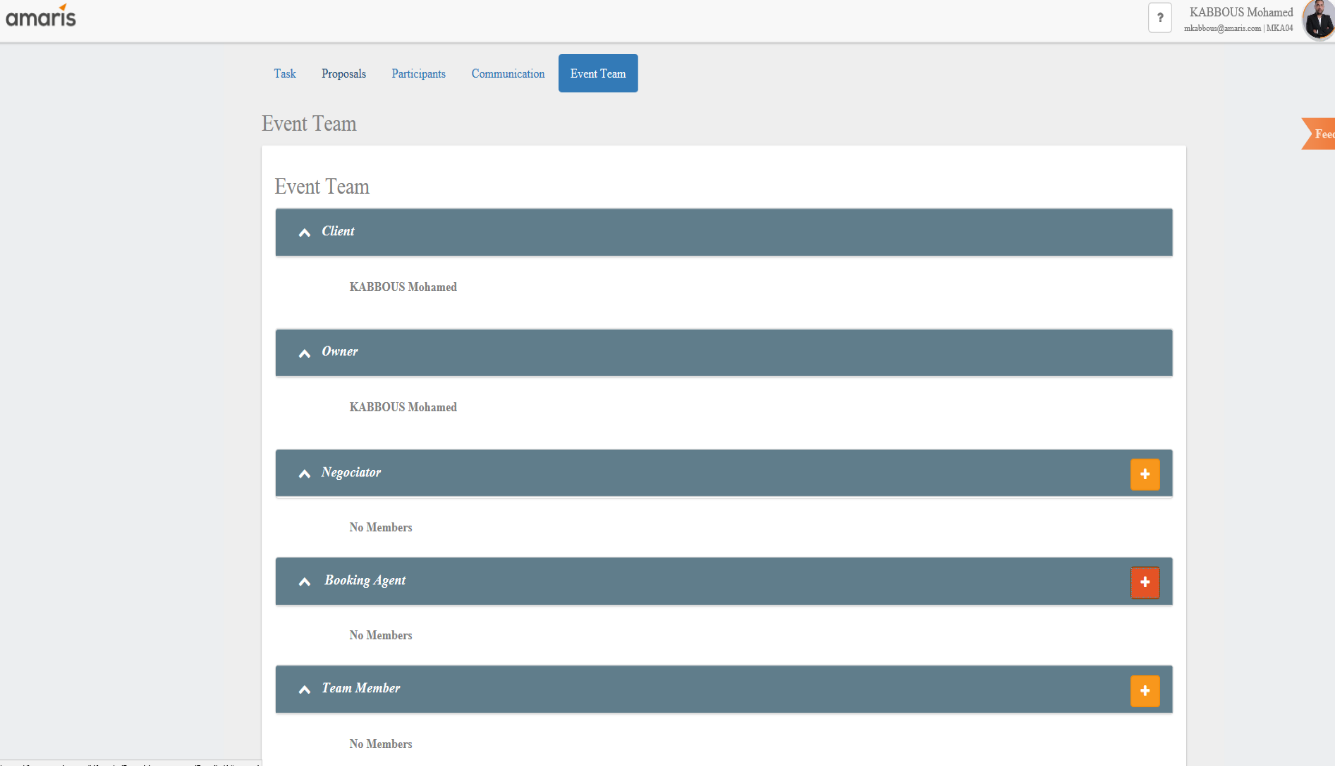


Figure 31 Assignation des différents membres (BookingAgent/Négociator)

Une fois qu’il clique sur le bouton (+) une fenêtre s’affiche contant les mêmes informations que celle lors d’assignation d’un Owner à un détail près ; celui de la « Current Event Team » qui contient des informations à l’équipe organisatrice avec le rôle de chacun.

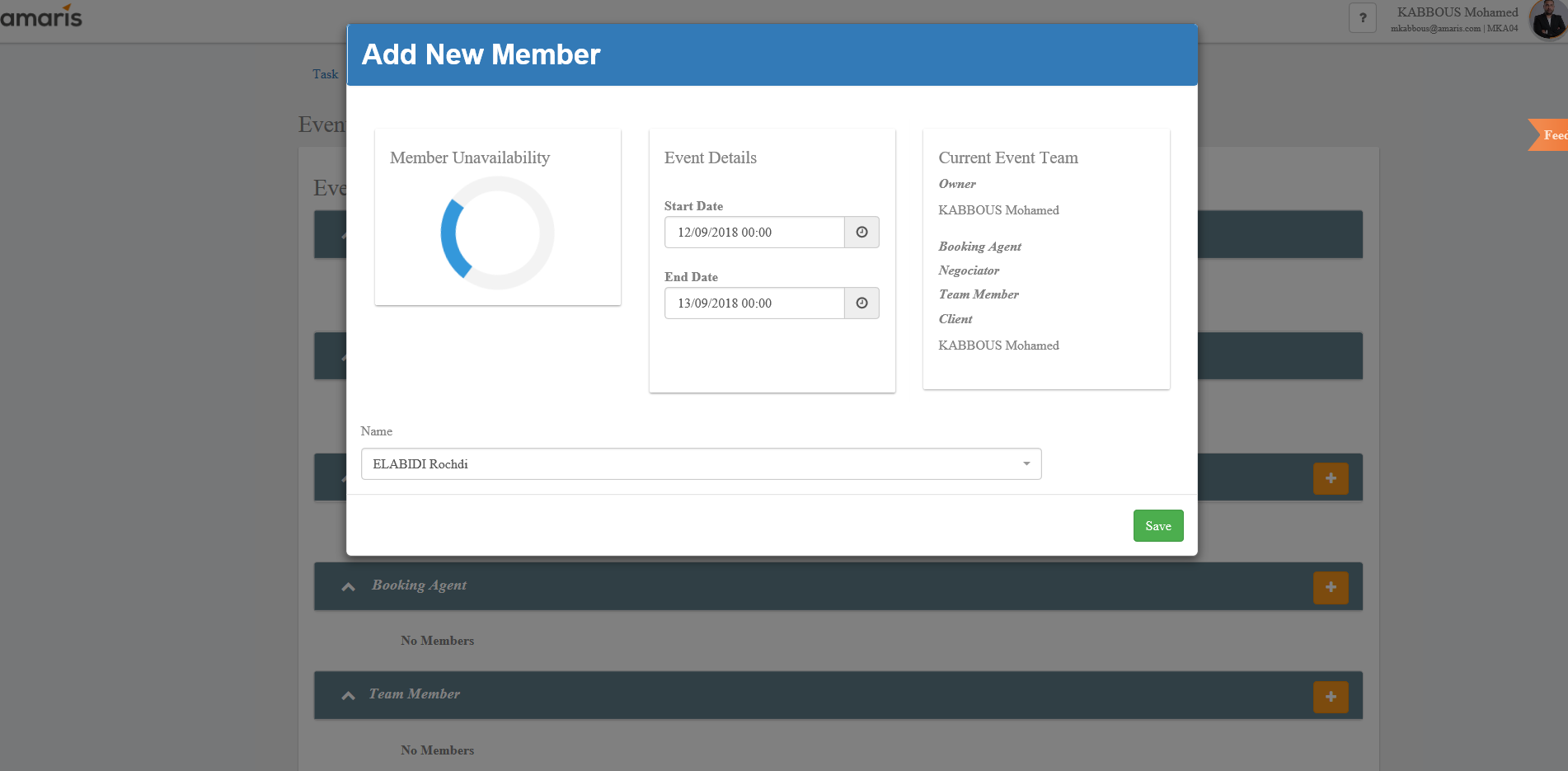


Figure 32 Interface du choix d'assignation

### Scénario invitation des participants

S’il veut inviter des personnes pour assister à cet événement le Requestor peut rajouter des participants en se rendant à la rubrique dédiée.

En effet une rubrique intitulée « Participants » nous permet d’avoir une interface qui contient la liste de tous les employés ainsi que la possibilité de recherche et aussi devant chaque employé nous avons un switch bouton si on l’active l’invitation sera envoyée si on le désactive l’invitation sera annulée ou bien il n’est pas encore invité

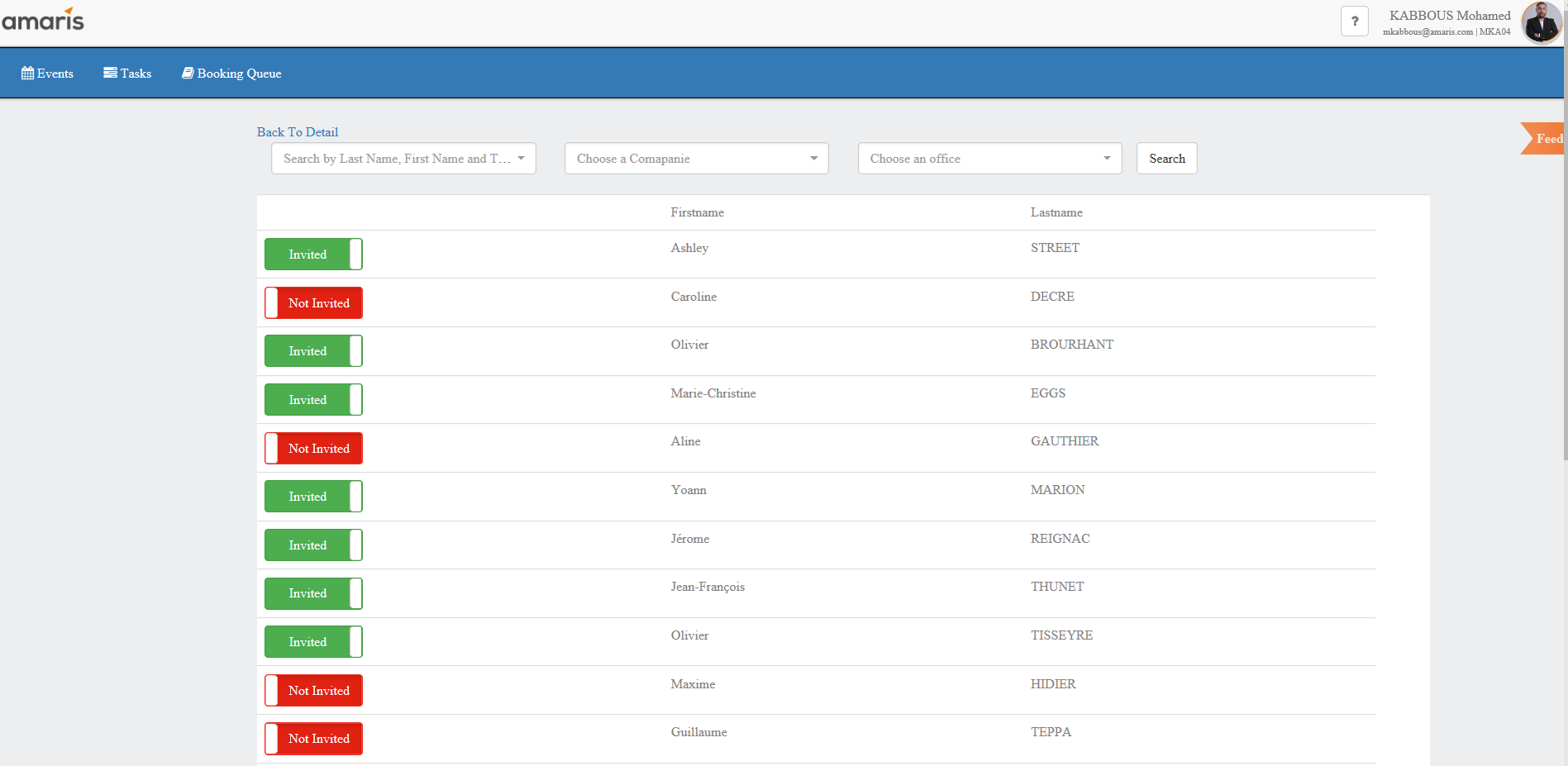


Figure 33 Interface d'invitation des participants

Ce dernier, a la possibilité aussi de consulter la liste des personnes qu’il a déjà inviteées, s’il veut faire une recherche un espace est dédiée pour la cause, et s’il veut en supprimer quelqu’un (annuler une invitation) un bouton qui contient une icône d’une corbeille est mis à disposition pour ladite cause.

Nous pouvons le constater à travers la figure suivante :

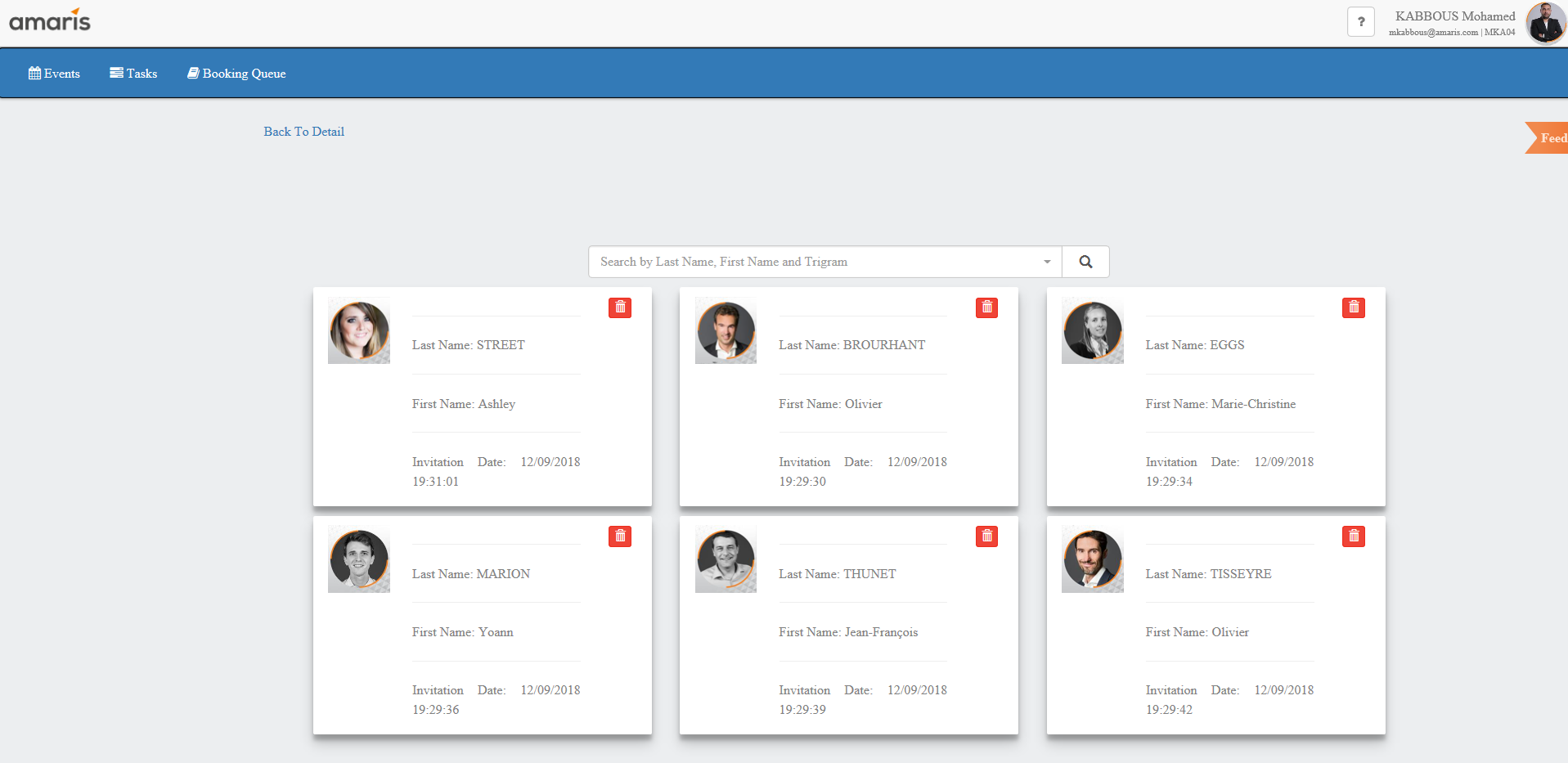


Figure 34 Interface des listes des participants

Conclusion

Dans ce dernier chapitre consacré à la réalisation proprement dite et à ses outils, méthodes et techniques, nous avons tout d’abord présenté l’environnement logiciel employé dans le développement de notre application, puis nous avons illustré par un exemple de déroulement du processus à travers des captures d’écrans.

# Annexe

## La plateforme Android

### Présentation

Android est un système d’exploitation open source, basé essentiellement sur Java, Il devient très rapidement célèbre dans le marché des téléphones mobiles qu’on appelle désormais Smartphone (les téléphones intelligents). Il doit cette célébrité au fameux moteur de recherche Google qui est son créateur avec une trentaine d’autres partenaires.

Dédié au grand public ; utilisateurs des Smartphones, tablettes tactiles, SmartWatches, Il vise à détrôner le roi du secteur IOS installé sur des Smartphones de la prestigieuse société Apple.

Géo localisation, appel des services distants, stockage des données en local et autre lecteur mp3, Android propose essentiellement une API complète ; en effet il apporte un vrai modèle architectural fondateur et original à base de composants et d’envoi des messages.

### Architecture de l’Android

Pour bien comprendre la plateforme Android, nous devons détailler l’architecture de cet OS. C’est pour cela qu’on a décidé de faire un schéma explicatif représenter par la figure 21.

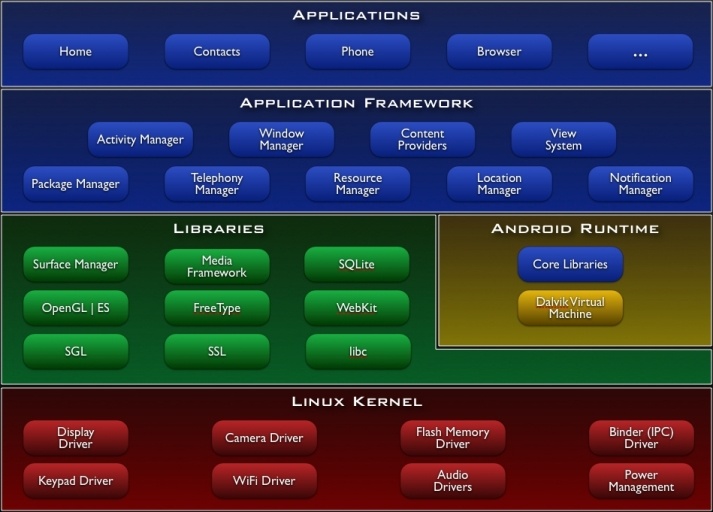


Figure 35Architecture Android

* **Linux Kernel**

Android est basé essentiellement sur le noyau Linux kernel. Ce dernier possède un système de gestion mémoire et de processus reconnu par ses performances et sa stabilité. Il joue le rôle d’intermédiaire entre le matériel et les programmes grâce à son système. Il permet également le partage librairies entre différent processus, le chargement et le déchargement de modules à chaud.

* **Librairies :**

Android fournit un ensemble de librairies C/C++ utilisées par différents composants du système dans le but d’accélérer le développement d’applications. Ces fonctionnalités sont rendues à la disposition des développeurs grâce au Framework d’application d’Android. On site parmi les librairies : librairie C standard, moteurs d’affichage 2D et 3D, SQLite, rendu des polices de caractères etc.

* **Android Runtime :**

Android rassemble les librairies qui fournissent la majorité des fonctionnalités des librairies standards de java. Chacune des applications Android s’exécute dans un processus avec sa propre instance de Dalvik. En effet cette dernière est une machine virtuelle Java spécialement optimisée et conçue pour Android qui permet d'exécuter le même programme sur une grande variété d'appareils, indépendamment de leurs caractéristiques techniques.

#### **Application Framework**

La première phase du travail avec l’environnement Android consiste à appréhender le SDK, l’architecture et le développement d’une application ainsi que son déploiement sur un terminal embarquant Android.

# Conclusion générale

Ce document a été rédigé au terme du projet de fin d’études réalisé au sein de l’entreprise AMARIS, il s’agit de concevoir et de développer un produit de gestion d’événement et de tâche web et mobile (Android) qui résoudrait et superviserait l’ensemble des ressources du groupe.

La première phase de ce rapport a été consacrée à la présentation de l’organisme d’accueil, ainsi que son organigramme.

La deuxième phase quant à elle, a été consacrée à la présentation de l’état de l’art, ainsi que le projet à implémenter. Ces deux parties ont été suivies d’un passage en revue des solutions existantes et de leurs limites, ce qui nous a permis de poser la première pierre de l’édifice en proposant notre propre solution.

La troisième phase quant à elle consistait à dégager les besoins fonctionnels, et par là même les besoins non fonctionnels suivant la méthodologie de développement SCRUM. Cela nous a permis d’établir les diagrammes de cas d’utilisation qui nous ont servi dans la phase suivante : la conception qui nous a permis d’aller plus en profondeur dans la conception du projet avec un diagramme de cas d’utilisation, le diagramme de classe d’analyse et le diagramme se séquence système.

La quatrième phase, celle-ci nous a permis de spécifier mieux notre besoin, par le diagramme de classe de conception, le différent diagramme de séquence objet ainsi que la spécification des différents acteurs qui interviennent au niveau de l’application.

Enfin, dans la phase de réalisation nous avons présenté les outils et technologies utilisés pour la réalisation de ce travail par les interfaces de notre application.

Au final, il est important de souligner que ce projet a atteint les objectifs fixés initialement, et au-delà du sentiment de satisfaction qui s’en suit, il m’a permis d’appliquer mes connaissances théoriques dans un contexte professionnel, de bénéficier de nouvelles connaissances venues compléter celles acquises tout au long de notre formation, mais surtout m’a permis de me rapprocher du monde professionnel.

Cependant, ce travail, comme toute œuvre humaine, n’est certainement pas exempt d’imperfections. De ce fait, lui apporter quelques améliorations fera de cette application un outil d’encaissement incontournable.

Les perspectives d’améliorations et de développement de ce travail peuvent être multiples.

Un module consacré aux statistiques et aux divers rapports (de fin de journée, pour une semaine, un mois, une saison ou une période quelconque) donnera au groupe plus de visibilité et une aide certaine à la prise de décision.

# 

# Webographie

3**. Architecture des applications mobiles**,

<https://fr.slideshare.net/LiliaSfaxi/chp3-architecture-logicielle-des-applications-mobiles>

4. **Developer Guides**, Android Developers

<https://developer.android.com/guide/>

5. **UML basics**, IBM

<https://www.ibm.com/developerworks/rational/library/3101.html>

13. **Host ASP.NET Core on Windows IIS**

<https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/?view=aspnetcore-2.1&tabs=aspnetcore2x>