

**Institut Supérieur de l'Électronique et du
Numérique**

Tél. : +33 (0)2.98.03.84.00
Fax : +33 (0)2.98.03.84.10
20, rue Cuirassé Bretagne
CS 42807 - 29228 BREST Cedex 2 - **FRANCE**

PROJET M1
ANNÉE SCOLAIRE 2015/2016

Gestion des adhérents d'une association
<https://assaucisson.herokuapp.com>



Auteurs :
Etienne JÉZÉQUEL
Mathieu GAUTHIER

Sujet proposé par :
Isabelle LÉONARD

Remerciements

Nous remercions l'ISEN de nous avoir permis de réaliser ce projet. Nous remercions également Mme Isabelle Leonard d'avoir proposé le sujet et d'avoir su jouer le rôle de cliente durant toute la durée du projet.

Nous remercions enfin GitHub de nous avoir offert un dépôt privé et Heroku pour nous avoir permis d'héberger le projet gratuitement sur un serveur distant.

Résumé

Le but de ce projet est de créer une application permettant de gérer plus facilement les adhérents d'une association. Son deuxième intérêt est de passer d'une gestion d'association au format papier ou excel à celui du numérique.

L'objectif est d'appréhender le sujet de manière professionnelle en proposant des solutions qui permettront de faire évoluer le projet dans le futur.

En d'autres termes, il s'agit d'effectuer une migration des flux de données (informations des membres de l'association) vers une base et de réaliser une transformation digitale au sein de l'association de la cliente Mme Isabelle Léonard.

Table des matières

1	Glossaire	1
2	Introduction	4
3	Présentation du projet	5
1	Cahier des charges initial	5
2	Les solutions proposées	5
2.1	Première solution : Application native	5
2.2	Deuxième solution : Application Web	6
3	Conclusion	6
4	Gestion de projet	7
1	Analyse fonctionnelle du besoin	7
1.1	Bête à cornes	7
1.2	Diagramme pieuvre	8
1.3	Organigramme fonctionnel	8
2	Cahier des charges fonctionnel	11
2.1	Description fonctionnelle	11
2.2	Description graphique et ergonomique	12
3	Planification	18
4	Plans de tests	20
4.1	Modifications des objectifs	22
5	Méthodologie	24
6	Outils de développement et de gestion	25
6.1	Wrike	25
6.2	Tableau d'avancement	25
6.3	GitHub	26
5	Partie technique	27
1	Choix technologiques	27
1.1	Partie cliente	27
1.2	Partie serveur	30
1.3	Pourquoi de tels choix technologiques ?	32
2	Environnement de travail	33
3	Exploitation des résultats	35
3.1	Résultat de l'application finale	35
3.2	Futures améliorations envisageables	35
6	Conclusion	37
7	Annexes	38

Table des figures

4.1	Bête à cornes	7
4.2	Diagramme pieuvre	8
4.3	Organigramme fonctionnel	10
4.4	Tableau des fonctions principales	11
4.5	Tableau des fonctions de contraintes	12
4.6	Maquette fonctionnelle représentant les éléments hors-connexion de l'utilisateur	14
4.7	Tableau de bord et liste des adhérents	15
4.8	Gestion des adhérents et des groupes (Administrateur)	16
4.9	Gestion des activités et participations des membres	17
4.10	Planning prévisionnel	18
4.11	Tableau des plans de tests (Extrait)	22
4.12	Nouvelle page de connexion (Maquette fonctionnelle)	23
4.13	Nouvelle page d'inscription (Maquette fonctionnelle)	23
4.14	Schéma de la méthode agile	25
5.1	Schéma explicatif du fonctionnement de Angular	27
5.2	Schéma explicatif du fonctionnement de Redux	29
5.3	Logo Firebase	31
5.4	Schéma JSON généré par Firebase	32
5.5	Schéma explicatif du fonctionnement de Node JS	33
5.6	Résultat final de l'application Web	35
7.1	Compte-rendu de réunion du 15/12/2015	38
7.2	Compte-rendu de réunion du 05/01/2016	39
7.3	Compte-rendu de réunion du 11/01/2016	40

Chapitre 1

Glossaire

Termes techniques

Glossaire concernant les technologies abordées dans le rapport.

AngularJS

AngularJS est un framework JavaScript libre et open-source développé par Google à destination du WEB. Ce framework se base sur l'extension du langage html en ajoutant de nouvelles balises et de nouveaux attributs. Angular utilise des modèles appelés « *scopes* » et des contrôleurs qui permettent de définir des actions en code javascript impératif. Avec ce framework, la partie HTML représente la partie vue du modèle MVC et les contrôleurs, la partie contrôle du modèle MVC.

HTML/CSS

L'Hypertext Markup Language (html) est le format de données conçu pour représenter les pages web. C'est un langage de balisage. HTML permet également de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie. Le HTML est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade : CSS).

Les feuilles de style en cascade, appelées **CSS** (Cascading Style Sheets), forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web.

Javascript

JavaScript est un langage de programmation de scripts utilisé dans les pages web mais aussi pour les serveurs (par exemple NodeJs). C'est un langage impératif qui permet principalement d'ajouter du dynamisme à un site web.

SQL

SQL (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données. En plus de la manipulation des données, sql permet la définition des données afin de créer et de modifier l'organisation des données dans la base de données

Apache

Le logiciel Apache HTTP Server est un serveur HTTP libre. Il a été et il est maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire. Un serveur http est un serveur Web. Il se sert des requêtes respectant le protocole de communication client serveur Hypertext Transfer Protocol (HTTP) afin de communiquer.

PHP

Hypertext Preprocessor est un langage de programmation libre, utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété. C'est un langage que seuls les serveurs comprennent et qui permet de rendre votre site dynamique. PHP est un langage impératif orienté objet.

Serveur

Un serveur est un dispositif informatique matériel (concrètement un ordinateur) ou logiciel qui offre des services. Les services les plus courants sont :

- l'accès aux informations du World Wide Web (CAS D'APACHE)
- le courrier électronique
- le partage d'imprimantes
- le stockage en base de données

Un serveur fonctionne généralement en permanence. Il doit répondre automatiquement à des requêtes provenant d'autres dispositifs (les clients).

Responsive Design

C'est un procédé relativement récent qui consiste à adapter l'affichage du site en fonction du type d'appareil utilisé pour la consultation. Il existe de nombreux outils permettant d'assurer une bonne lisibilité et ergonomie, que le visiteur soit sur un ordinateur, un mobile ou une tablette.

Scalable

La scalabilité désigne la capacité d'un produit à s'adapter à une montée en charge. Même avec de fortes demandes, une application scalable peut maintenir ses performances et ses fonctionnalités.

Base de données

Une base de données est un outil permettant de stocker et de retrouver des données. Dans la majorité des cas, ces informations sont structurées. La base est localisée dans un même lieu et sur un même support et est généralement informatisée.

Client

Le client est l'utilisateur de l'application.

Front-End

La partie Front-End d'une application correspond à la partie cliente de celle-ci.

Back-End

La partie Back-End d'une application correspond à la partie serveur de celle-ci.

UI

UI vient de l'anglais “User Interface” qui signifie interface utilisateur. Cela correspond à la partie cliente (le Front-end) d'une application.

SaaS

Le logiciel en tant que service ou software as a service (SaaS) sont des logiciels installés sur des serveurs distants plutôt que sur la machine de l'utilisateur.

HTTP

L'HyperText Transfer Protocol est un protocole de communication client serveur développé pour le World Wide Web (WWW).

Framework

Un Framework est un ensemble de composants logiciels destiné à la conception d'architecture logicielle

Chapitre 2

Introduction

Durant la première année de Master à l'ISEN, des projets sont distribués aux étudiants de manière aléatoire. Le suivi de ces projets est réalisé par un professeur référent qui joue alors le rôle de client. Par conséquent, les projets se déroulent de la même manière que ceux réalisés dans une entreprise.

Cela a pour but d'évaluer les capacités d'adaptation des étudiants face à un environnement professionnel. La durée de chaque projet est limitée dans le second semestre de la première année de Master.

Ce rapport sera présenté en trois parties principales. La première partie sera la présentation du sujet argumentée d'une analyse fonctionnelle et d'un cahier des charges. La deuxième partie sera l'explication des choix technologiques et l'environnement de travail. La troisième partie présentera les problèmes survenus pendant les phases de développement et les solutions proposées aux clients pour les contourner ou les résoudre. Et enfin, dans une conclusion, une présentation du travail effectuée sera dévoilée.

Chapitre 3

Présentation du projet

1 Cahier des charges initial

Voir les annexes => Compte-rendu de réunion 15/12/15

La cliente, Mme Isabelle Léonard, est gérante d'une association brestoise et dans cette dernière, le procédé d'inscription des membres est redondant et difficile à gérer. En effet, les adhérents doivent remplir des fiches contenant un certain nombre d'informations (adresses, numéros de téléphone, mails, dates de naissance...).

L'objectif du projet est de créer une base de données et l'application intuitive de gestion des adhérents afin d'éviter cette tâche fastidieuse qu'est le traitement des fiches d'inscriptions. Il sera envisageable d'ajouter des fonctionnalités non demandées mais permettant d'améliorer l'expérience de l'utilisateur.

Exigences fonctionnelles :

- Ajouter de nouveaux adhérents et remplir les champs associés
- Supprimer des adhérents (mais pouvoir conserver les informations au cas où ils reviendraient dans le futur)
- Modifier les coordonnées des adhérents
- Créer des groupes
- Exporter tout ou seulement certains champs dans un fichier PDF. Par exemple, exporter un annuaire, une liste des adhérents, un tableau des anniversaires (dans l'ordre de l'année)
- Il doit être possible d'envoyer des mails à tous les adhérents ou seulement à une partie des adhérents
- Le produit doit pouvoir être utilisé par n'importe qui possédant un ordinateur ("clé en main").

2 Les solutions proposées

2.1 Première solution : Application native

Le logiciel de gestion en natif est une des solutions envisageables pour un tel projet. En effet, on retrouve de nombreux avantages pour ce genre de technologie :

- gestion du hors-ligne,
- très bonne performance,
- optimisation du contenu (navigation plus intuitive et plus adaptée au support)

Certains soucis pourront être à déplorer selon le type d'utilisateurs et les cas d'utilisation. Il est facile de constater que le logiciel de gestion est un client lourd c'est-à-dire qu'il nécessite une installation au préalable avant utilisation. De plus, la modularité d'un tel produit est difficilement envisageable car elle demande à l'utilisateur d'effectuer une mise à jour du produit en téléchargeant un nouvel

exécutable. Un autre problème redondant est celui de la compatibilité du produit. L’application native n’est pas forcément compatible avec tous les appareils (IOS, Android, Windows...).

2.2 Deuxième solution : Application Web

L’application Web est un logiciel applicatif hébergé sur un serveur et accessible par l’intermédiaire d’un navigateur (Mozilla Firefox, Edge, Google Chrome, Safari...).

Contrairement à un logiciel classique (comme expliqué précédemment), les applications web n’ont pas besoin d’être installées sur l’ordinateur de l’utilisateur. Il lui suffit de se connecter à l’application via son navigateur. L’objectif de ce type de produit est de pouvoir offrir à l’utilisateur une expérience et des fonctionnalités équivalentes aux logiciels classiques, en utilisant de technologies de développement identiques à celles employées habituellement dans la création des sites internet.

Les applications web offrent de nombreux avantages telles que la compatibilité avec tous les navigateurs modernes, l’adaptation à de plus faibles résolutions d’écran, le stockage de fichiers simplifié et la possibilité d’ajouter des fonctionnalités hors-ligne.

3 Conclusion

Après une réunion de travail avec la cliente, les atouts de l’application Web ont pris le dessus par rapport à ceux de l’application native. En effet, que ce soit d’un point de vue développeur ou utilisateur, les Web Apps disposent d’un intérêt sur le plan du temps de développement et sur la compatibilité entre plusieurs supports. Mme Isabelle Léonard, durant la réunion, soulignait le fait que l’application devait être adaptée à un public plus ou moins âgé. Effectuer un portage sur différents supports (certains pouvant être obsolètes) paraît très fastidieux et chronophage.

Dans ce cas, il est nécessaire d’adapter le choix de la Web App à la gestion de projet utilisée. De plus, une autre question s’est posée durant la réunion. Est-il nécessaire de porter l’application Web à un usage généralisé ? En d’autres termes, créer une Web App ayant la capacité de créer et gérer une multitude d’associations permettrait d’améliorer l’expérience utilisateur et d’augmenter l’adaptabilité du produit final. Un tel choix aura forcément des conséquences en matière de choix technologiques et de gestion de projet (gestion du temps et méthodologie employée).

Chapitre 4

Gestion de projet

1 Analyse fonctionnelle du besoin

Afin d'établir les axes de développement du projet, il est nécessaire d'analyser le besoin du client d'un point de vue fonctionnel c'est-à-dire de définir précisément chaque fonctionnalité et d'attribuer ces dernières au bon environnement.

1.1 Bête à cornes

L'étude du besoin s'articule sur plusieurs éléments de la gestion de projet qui permettent de délimiter le périmètre du projet ainsi que les fonctions principales et de contraintes. Afin de définir la cible de l'application, une bête à cornes est utilisée. Il est, en effet, nécessaire d'expliciter simplement le besoin primaire, c'est-à-dire l'exigence principale. Le but de ce schéma doit être de satisfaire un besoin exprimé par l'utilisateur.

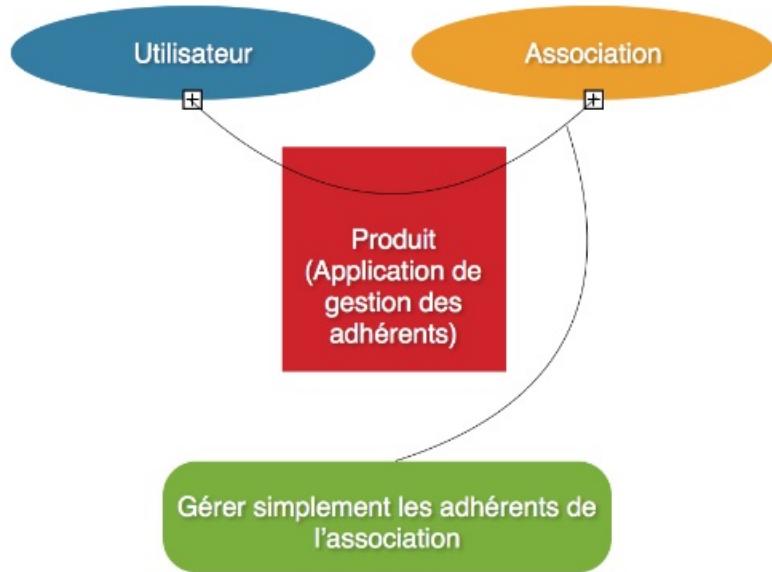


FIGURE 4.1 – Bête à cornes

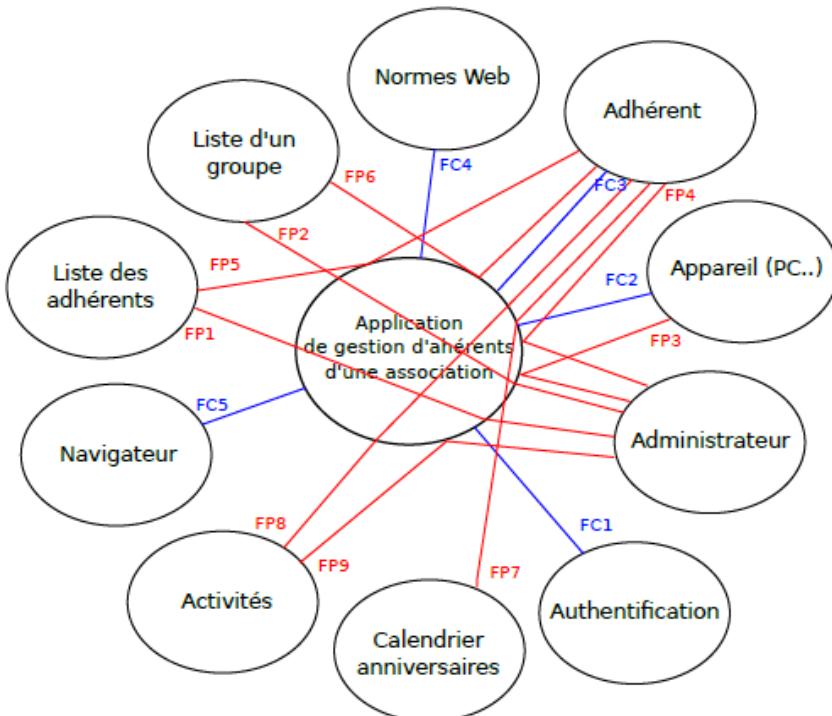
L'usage du produit doit générer des fonctions de services que la bête à cornes permet d'identifier et de caractériser. Dans ce cas, l'application web est destinée aux associations. Elle a pour but de simplifier la gestion des adhérents de celle-ci en offrant un outil simple d'utilisation et accessible par

tous (compatibilité et intuitivité).

La visée de l'application ne se concentre pas essentiellement sur un seul type d'association. L'objectif est de créer un outil pouvant créer et gérer plusieurs associations.

1.2 Diagramme pieuvre

On délimite le périmètre du projet selon les éléments du milieu extérieur qui agissent sur le produit. On utilise pour cela un diagramme pieuvre. Celui-ci permet alors de déterminer les fonctions principales et les contraintes du produit. Dans le cas ci-dessous, on retrouve les liens entre les éléments du milieu extérieur qui forment en rouge les fonctions principales (FP) et en bleu les contraintes (FC).



- FP1 : Un administrateur peut ajouter, modifier, supprimer des adhérents et en afficher la liste
- FP2 : Un administrateur peut ajouter, modifier, supprimer des groupes et en afficher la liste
- FP3 : Un administrateur peut exporter des informations sur l'application vers son appareil
- FP4 : Un administrateur peut envoyer une campagne de mails aux adhérents
- FP5 : Un adhérent peut afficher la liste des adhérents
- FP6 : Un adhérent peut afficher la liste d'un groupe
- FP7 : Un administrateur peut ajouter, modifier, supprimer des activités et en afficher le planning
- FP8 : Un adhérent peut afficher le calendrier des anniversaires
- FP9 : Un adhérent peut gérer sa présence à une activité et en afficher le planning
- FC1 : Il est nécessaire de se connecter à un système d'authentification
- FC2 : L'application doit être adaptée à tous les appareils (Mobile, tablette..)
- FC3 : L'application doit être simple d'utilisation pour les adhérents
- FC4 : L'application doit respecter les normes W3C (Web)
- FC5 : L'application doit être compatible avec les navigateurs les plus utilisés

FIGURE 4.2 – Diagramme pieuvre

1.3 Organigramme fonctionnel

L'organigramme fonctionnel permet de clarifier les différents scénarios d'accès à l'application. Il s'agit d'une représentation schématique des liens fonctionnels entre les fonctions du produit. On retrouve toutes les fonctions principales du diagramme pieuvre dans cet organigramme.

Dans le cas du projet, il est nécessaire de différencier les droits de chaque membre. Cette différenciation s'effectue lors de l'identification de l'utilisateur par l'intermédiaire d'un mot de passe et d'un identifiant (représenté par l'adresse mail de l'utilisateur). Ainsi, en fonction du rang du membre, l'affichage et les fonctionnalités sont différents.

Exemple : *Un administrateur pourra supprimer un adhérent alors que ce dernier ne pourra que consulter les membres de l'association.*

Cet organigramme fonctionnel ne met en avant que la partie "Identification d'un utilisateur". L'inscription s'effectue de manière différente. Une explication sera détaillée dans les prochaines sections.

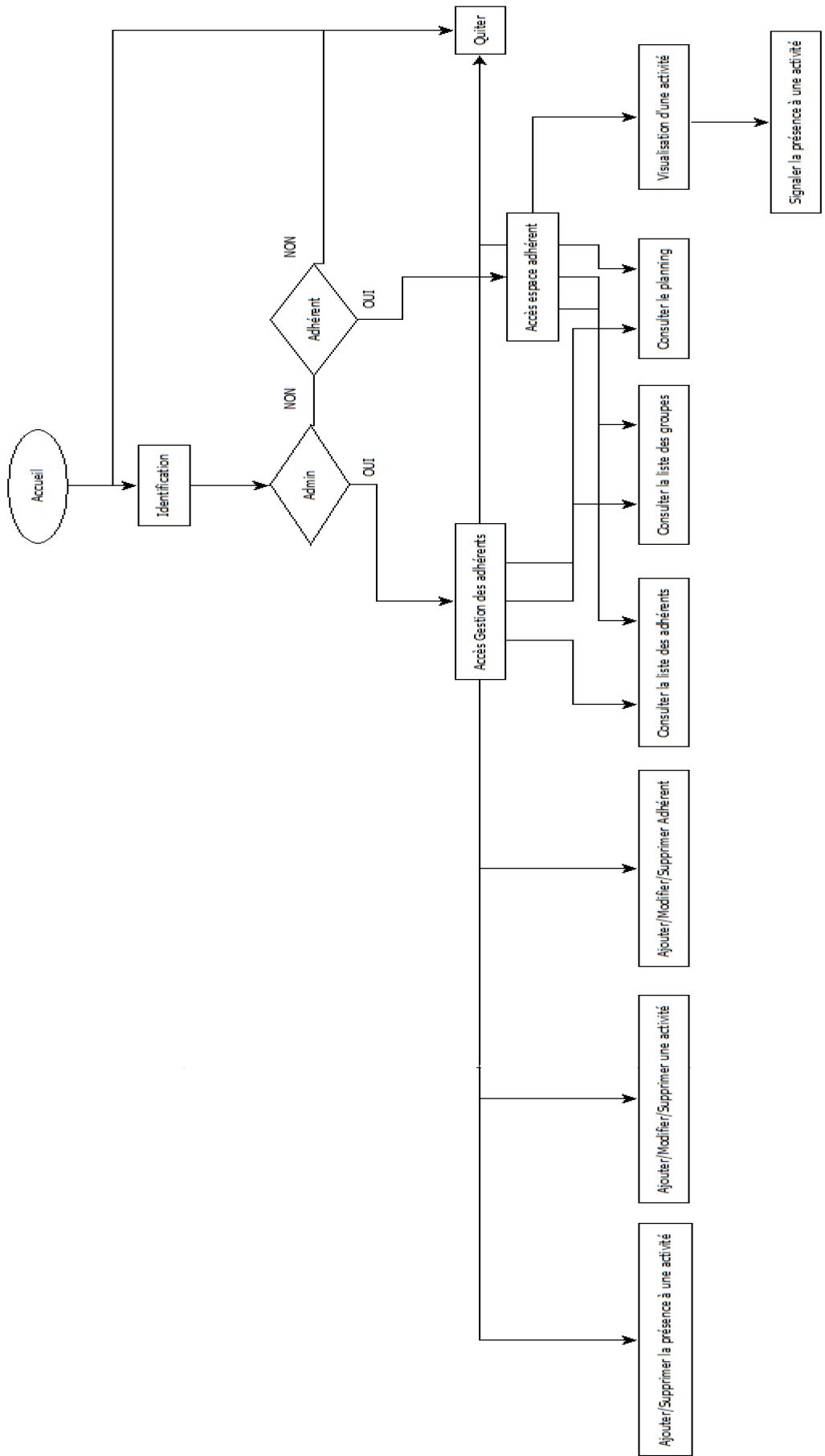


FIGURE 4.3 – Organigramme fonctionnel

2 Cahier des charges fonctionnel

2.1 Description fonctionnelle

Dans le cahier des charges fonctionnel, toutes les fonctions principales et de contraintes sont classées dans un tableau. Pour chaque fonction, il suffit de définir :

- Les critères d'application
- Les niveaux d'exigence
- Les niveaux de flexibilité

Le **critère d'application** permet d'indiquer dans quelles conditions la fonction (principale ou de contrainte) est considérée comme réalisée.

Par exemple, dans le cas de la gestion des adhérents, cette fonction est réalisée si et seulement si les fonctionnalités permettant de modifier, ajouter et supprimer des adhérents sont complètement réalisées.

Le **niveau d'exigence** représente la valeur qui est associée au critère d'application. Cela peut être alphanumérique. *Dans le cas de la gestion des groupes (par exemple), le critère d'application est la modification, la suppression et l'ajout de groupe. On exige qu'il n'y ait pas de suppression en cascade c'est-à-dire que la suppression d'un groupe n'entraîne pas celle des membres appartenant à ce dernier.*

La **flexibilité** permet d'accorder une tolérance au niveau de réalisation d'une fonction. *Par exemple, dans le cas de la fonction de contrainte "Compatibilité avec les navigateurs les plus utilisés", on demande à l'application d'être compatible avec les anciennes versions IE (IE8) mais si les circonstances du projet ne le permettent pas, on peut ne transmettre la compatibilité qu'aux navigateurs modernes.*

Fonctions principales	Critère d'application	Niveau d'exigence	Flexibilité
FP1 : Gestion des adhérents	Modifier, ajouter et supprimer	S'effectue en temps réel	Aucune
FP2 : Gestion de groupes	Modifier, ajouter et supprimer	Pas de suppression en cascade	Aucune
FP3 : Exportation du contenu en PDF	Mise en page réussie	Qualité optimale sans perte d'information	Aucune
FP4 : Envois de mails	Intégration de liste de diffusion		Aucune
FP5 : Afficher la liste des adhérents	Afficher par ordre alphabétique	Aucun	Aucune
FP6 : Afficher la liste des groupes	Afficher par ordre alphabétique	Aucun	Aucune
FP7 : Afficher un calendrier d'anniversaires	Afficher l'année actuelle et gestion des anniversaires	Aucun	Aucune
FP8 : Gestion et affichage d'activités	Ajout et suppression des activités	Aucun	Aucune

FIGURE 4.4 – Tableau des fonctions principales

Fonction de contrainte	Critère d'application	Niveau d'exigence	Flexibilité
FC1 Système d'authentification	Sécurité par mot de passe aléatoire	Sécurité maximale	Aucune
FC2 Site responsive	Optimisation pour les téléphones ; tablette etc...	Optimisation maximale	360 x 640 pixels
FC3 Système d'authentification	Sécurité par mot de passe aléatoire	Sécurité maximale	Aucune
FC4 Respect des normes Web W3C	Aucune erreur de syntaxe	Aucun	Cette contrainte peut être abandonnée si de
FC5 Compatibilité avec les navigateurs les plus utilisés	Compatible avec les anciennes versions IE (IE8)	Affichage et fonctionnalités similaires sur la plupart des navigateurs	Navigateurs modernes seulement

FIGURE 4.5 – Tableau des fonctions de contraintes

2.2 Description graphique et ergonomique

Voir les annexes => Compte-rendu de réunion 05/01/16

Charte graphique

En matière de charte graphique, l'objectif est de s'appuyer au maximum sur celle de l'ISEN qui nous est fournie. L'administrateur d'une association pourra définir le code couleur du dashboard ainsi que modifier des éléments visuels tel que son logo. Cette fonctionnalité supplémentaire permet avant tout de personnaliser l'interface utilisateur de manière à donner une identité à l'association.

Design

L'intuitivité et l'expérience utilisateur sont au centre des préoccupations en matière de design. L'application doit rester très sobre, moderne et minimaliste. Ce choix est dû à l'adaptation aux adhérents des associations. L'application risque d'être confrontée à des utilisateurs de tout âge avec des compétences en informatique toutes différentes.

Maquettes (Mockup)

Les maquettes fonctionnelles ou wireframes servent à représenter de façon visuelle l'ensemble des éléments qui doivent figurer sur une page comme les titres, textes, images, boutons et même les liens. L'intérêt du maquettage fonctionnel est d'affiner la structure de l'application et de revoir l'organisation interne des pages pour créer la meilleure expérience utilisateur possible. De plus, avec un logiciel de wireframes, il est simple de composer des interfaces différentes selon la taille de l'écran.

- **Page d'accueil [Figure 2.6 (a)]** : Le concept de la page d'accueil est d'effectuer une présentation rapide de l'outil avec quelques éléments visuels percutants et des boutons accessibles facilement.
- **Tableau de bord [Figure 2.7 (a, b)]** : Un menu regroupant les principales fonctionnalités sera présent sur la droite avec le logo de l'association correspondante. Celui-ci sera défini par le

responsable. Dans la partie gauche, il est possible d'afficher les informations relatives à l'association comme des actualités, les prochains événements, etc...

- **Inscription [Figure 2.6 (c, d, e, f)]** : L'inscription est particulière comparée à d'autres sites classiques. En effet, elle s'articule sur trois étapes distinctes. La première correspond aux informations (Nom, prénom, e-mail, Date de naissance) relatives à l'administrateur de l'association. La seconde demande à l'utilisateur le nom de l'association ainsi que sa date de création. Et enfin, la troisième étape regroupe les membres de l'association en ajoutant leurs adresses mail. Une fois, l'inscription terminé, les informations sont entrées en base de données et un mail contenant un mot de passe temporaire et aléatoire est envoyé vers les membres de l'association.
- **Connexion [Figure 2.6 (b)]** : L'utilisateur se connecte très facilement à l'aide de son identifiant (adresse mail) et son mot de passe (celui du mail envoyé précédemment). Il sera possible après la première connexion de modifier ce mot de passe sur un onglet Profil.
- **Gestion des adhérents [Figure 2.7 (c, d, e, f)]** : La liste des adhérents (e) apparait sous forme d'un tableau trié par ordre alphabétique. Chaque ligne du tableau correspond à un membre (Nom, prénom, adresse mail). A l'extrémité de chaque ligne, des icônes symbolisant la suppression et la modification du membre sélectionné. Dans le cas d'un membre non administrateur, la liste des adhérents (f) apparait sous la même forme exceptés des icônes l'empêchant de pouvoir agir sur les adhérents.
- **Gestion des groupes [Figure 2.8 (d, e, f)]** : La liste des groupes est très similaire à celle des adhérents. Il est possible d'ajouter et supprimer facilement des membres d'un groupe depuis une fenêtre contenant la liste des membres. Chaque nom comporte sur la gauche une case à cocher. De manière évidente, une fois la case cochée, le membre est ajouté dans le groupe en question. Seuls les administrateurs de l'association peuvent accéder à la modification et l'ajout des groupes.
- **Calendrier [Figure 2.9 (a, b)]** : Il était suggéré et plus ergonomique de regrouper le calendrier des anniversaires et celui des activités. Les anniversaires seront automatiquement programmés sur le calendrier une fois que l'utilisateur aura modifié son profil comprenant son adresse postale, son numéro de téléphone, son nom, son prénom et sa date de naissance. Selon ses informations, il sera envisageable de déterminer son âge et d'ajouter son anniversaire chaque année. Les membres et/ou administrateurs pourront choisir de participer ou non à un événement. Il est possible de participer à un anniversaire comme à une activité organisée par l'association.

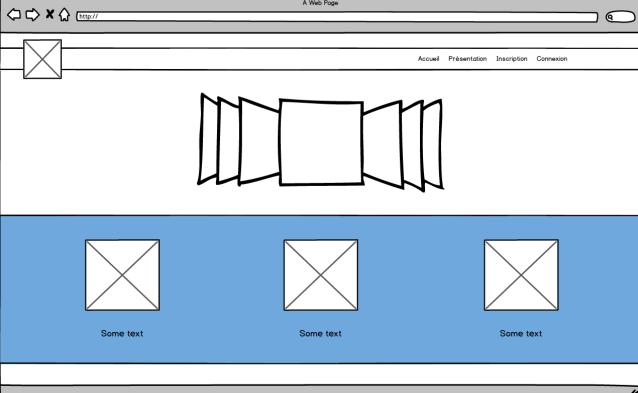
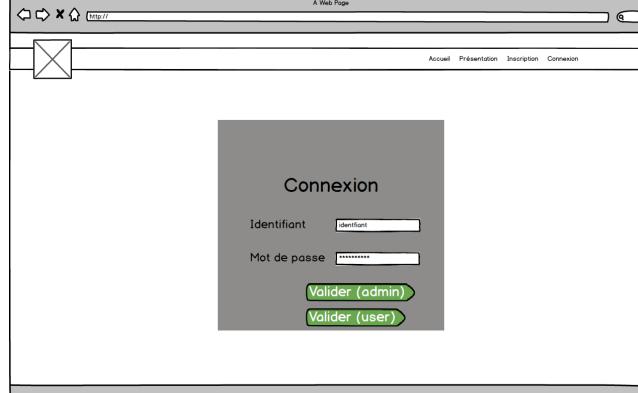
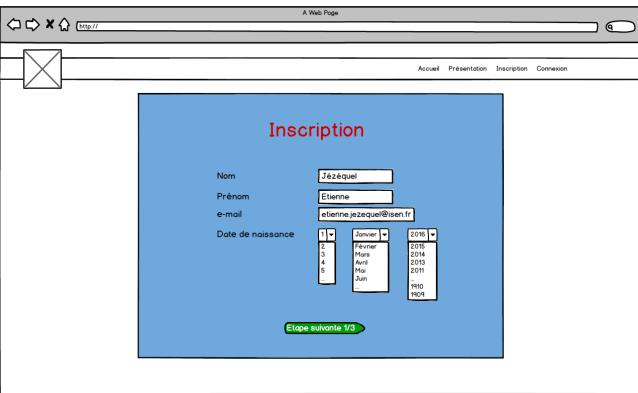
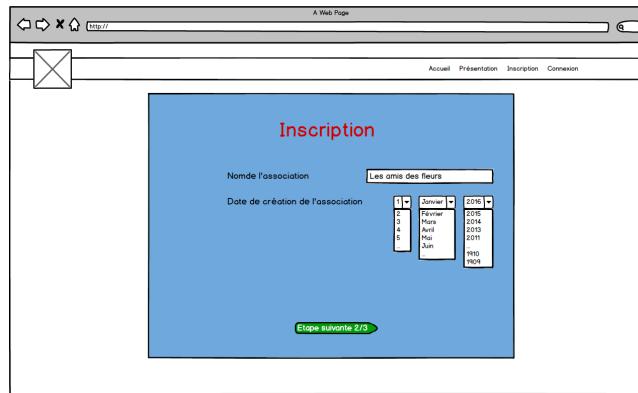
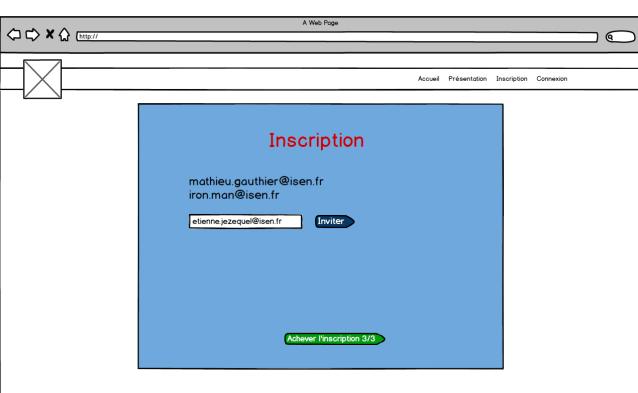
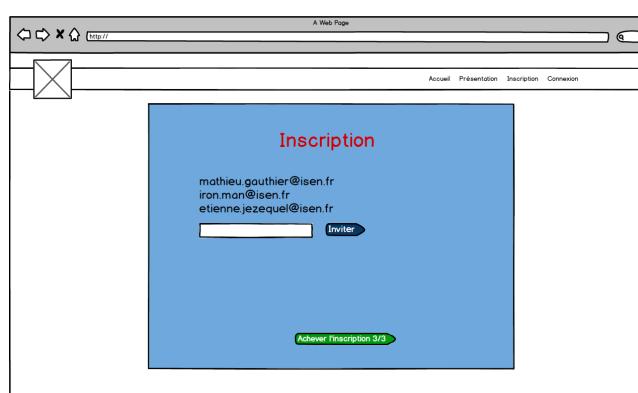
 <p>(a) Page d'accueil</p>	 <p>(b) Connexion</p>
 <p>(c) Première étape de l'inscription</p>	 <p>(d) Deuxième étape de l'inscription</p>
 <p>(e) Dernière étape de l'inscription</p>	 <p>(f) Ajout d'un membre dans l'association</p>

FIGURE 4.6 – Maquette fonctionnelle représentant les éléments hors-connexion de l'utilisateur

(a) Affichage du tableau de bord pour un administrateur

(b) Affichage du tableau de bord pour un membre

(c) Liste des adhérents (administrateur)

(d) Suppression d'un utilisateur

(e) Rechargement automatique de la liste une fois modifiée

(f) Liste des adhérents (membre)

FIGURE 4.7 – Tableau de bord et liste des adhérents

(a) Détail d'un membre sélectionné dans la liste des adhérents

(b) Modification des champs (adresse, nom, prénom, date de naissance...)

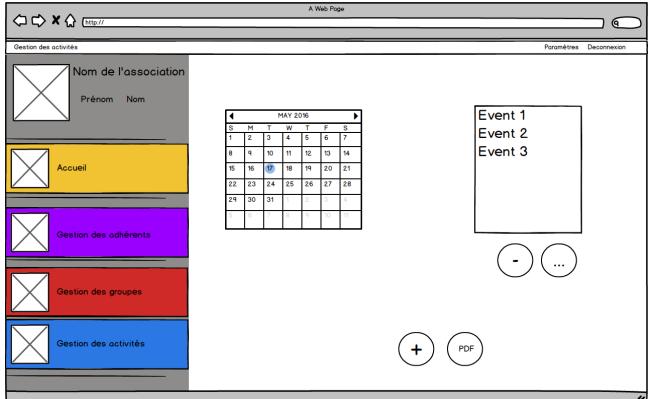
(c) Validation des modifications

(d) Liste des groupes (Administrateur)

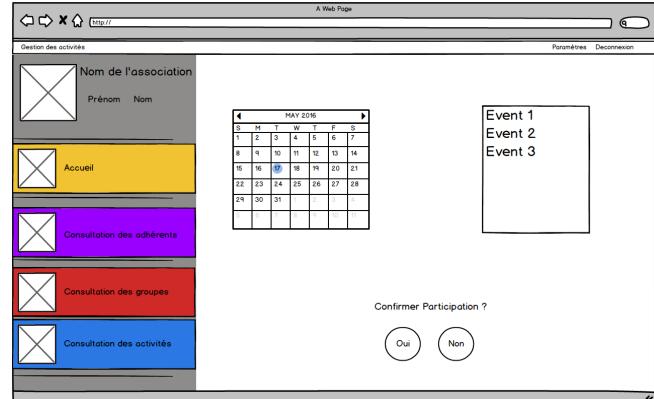
(e) Ajout ou suppression de membres dans un groupe

(f) Liste des groupes (membre)

FIGURE 4.8 – Gestion des adhérents et des groupes (Administrateur)



(a) Calendrier anniversaires et activités (Administrateur)



(b) Calendrier anniversaires et activités (Membre)

FIGURE 4.9 – Gestion des activités et participations des membres

3 Planification

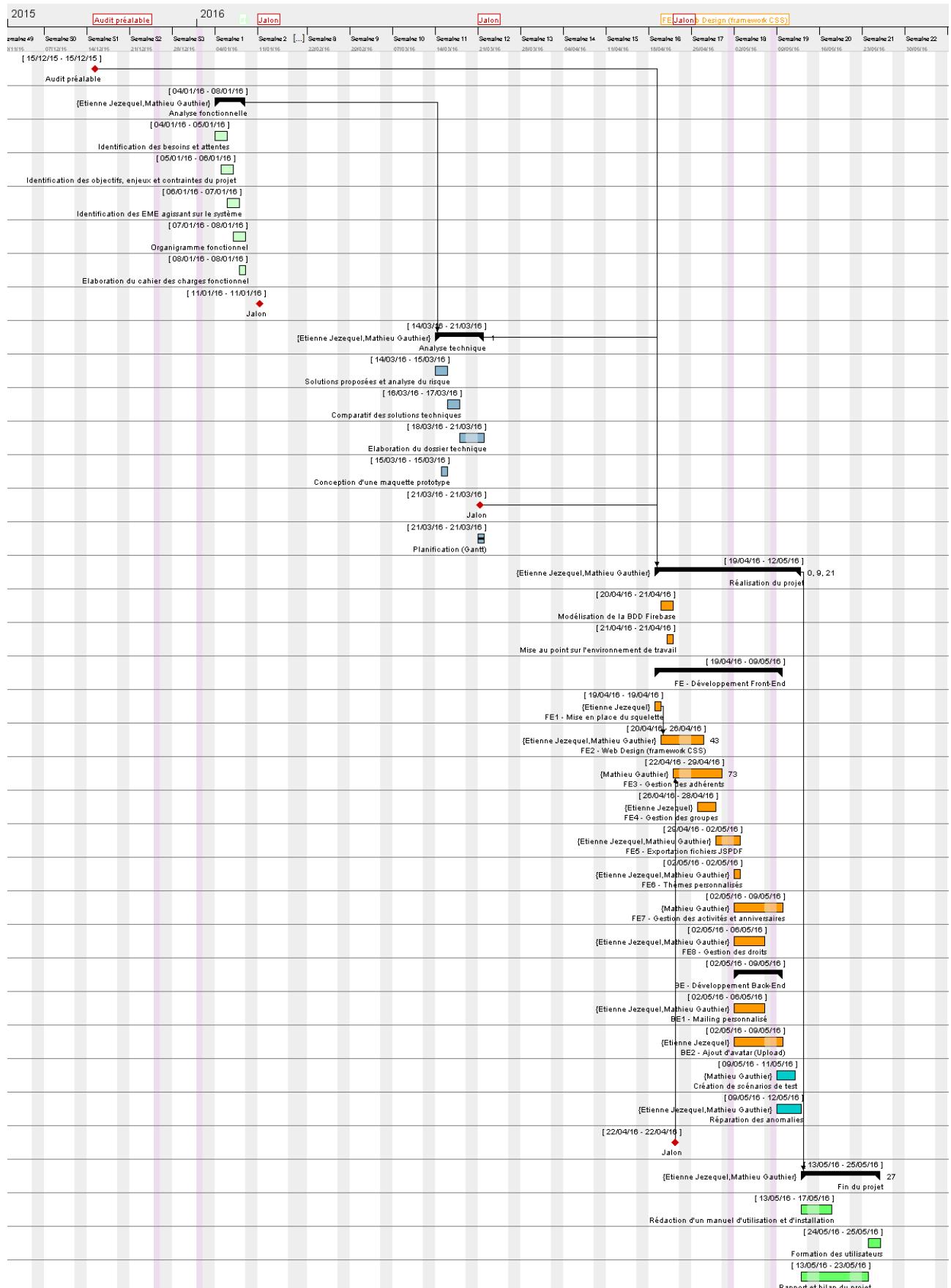


FIGURE 4.10 – Planning prévisionnel



La répartition et l'organisation ont abouti à la réalisation de ce premier diagramme de Gantt (ci-dessus) établi à l'aide du logiciel GanttProject. Les différentes étapes de la gestion de projet sont répertoriées sur ce diagramme.

L'avant projet regroupant toutes les phases d'analyses (fonctionnelle et technique) sont indispensables à la bonne compréhension du sujet initial et à l'élaboration d'un cahier des charges fonctionnel. Cela permet aussi d'anticiper les risques dans la réalisation et ainsi, de prévoir des phases de tests. Pour vérifier dans les meilleures conditions possibles l'adéquation du produit fini avec le cahier des charges initial, il sera nécessaire de préparer la phase de recette. Des plans de tests sont rédigés dès que la conception est validée auprès du client. Un plan de test se divise en deux lots :

1. Les tests fonctionnels

2. Les scénarios d'usage de l'application

Dans le premier cas, les tests ont pour but de vérifier la cohérence fonctionnelle de l'application Web c'est-à-dire que lorsque l'utilisateur effectue une action erronée ou incomplète, les conséquences de cette action doivent être cohérentes.

Par exemple : si un utilisateur entre un mauvais mot de passe lors de la phase de connexion au tableau de bord de l'association, il doit être renvoyé sur l'écran d'authentification et un message d'erreur est normalement affiché. Si ce n'est pas le cas, on remarque une anomalie.

Le second cas est différent car cette fois, les scénarios de test (ou d'usage) s'appliquent sur le produit fini dans des conditions réelles. Ils permettront de vérifier si l'application fait bien ce qu'elle a à faire en temps voulu. En effet, on va tester l'utilisabilité du produit final et constater s'il existe des erreurs de conception.

Il est important de définir précisément dans le planning les étapes de développement. Chaque élément concerné par un type de développement bénéficie d'un code (FE1¹, FE2, BE1²...). Ce dernier est réutilisé sur le tableau physique distribué au client et permet de repérer très rapidement les tâches à effectuer lors des stand-up et des réunions d'état d'avancement.

-
2. FE : Développement Front-End
 2. BE : Développement Back-End

4 Plans de tests

Numéros d'étape	Description de l'étape	Actions	Résultat attendu	Résultat obtenu
FE1	Mise en place du squelette de l'application	L'utilisateur se déplace entre les différentes pages (Connexion, Inscription...)	Transition fonctionnelle sans rechargement de la page	OK
FE3	Gestion des adhérents	L'administrateur supprime un membre	Destruction de l'utilisateur en BDD mais son compte n'est pas supprimé (il faut qu'il puisse se connecter dans les autres associations)	OK
FE3	Gestion des adhérents	L'administrateur modifie une ou plusieurs information(s) d'un membre	Modification dans la BDD et retour sur la liste des adhérents	OK
FE3	Gestion des adhérents	L'administrateur ajoute un membre	Envoi d'un mail explicatif au nouveau membre et ajout d'une information en BDD	OK
FE3	Gestion des adhérents	L'administrateur se supprime	Cette action est impossible	OK
FE4	Gestion des groupes	L'administrateur ajoute et supprime un membre d'un groupe en cochant la case correspondante	Modification d'un des attributs du membre en BDD	OK

Numéros d'étape	Description de l'étape	Actions	Résultat attendu	Résultat obtenu
FE5	Exportation du calendrier et du trombinoscope des adhérents	L'utilisateur télécharge un document PDF contenant un tableau s'adaptant automatiquement à la page	Téléchargement effectué et aucun changement en BDD	OK
FE6	Thèmes personnalisés	L'administrateur choisit un thème parmi 4 proposés	Modification de l'attribut "theme" en BDD avec le nom du thème sélectionné (Mise à jour en temps réel)	OK
FE7	Gestion des activités et anniversaires	L'utilisateur inscrit son anniversaire sur son profil	Ajout d'un événement spécial dans la BDD et le calendrier (se génère chaque année)	OK
FE7	Gestion des activités et anniversaires	L'utilisateur participe à un évènement (anniversaire ou activité)	Modifications en BDD et affichage de l'événement mis à jour avec les noms des participants	OK
FE7	Gestion des activités et anniversaires	L'administrateur ajoute un événement dans le calendrier	Ajout dans la BDD d'un événement et mise à jour de l'affichage en temps réel dans le calendrier	OK

FE8	Gestion des droits	Les utilisateurs essayent d'accéder aux pages Tableau de bord, Gestion des adhérents etc... alors qu'ils ne sont pas connectés	Redirection vers la page de connexion	OK
FE8	Gestion des droits	Les utilisateurs essayent de supprimer, modifier et ajouter des groupes (ou adhérents et événements) exceptés leurs anniversaires	Action impossible car l'affichage ne peut pas l'autoriser	OK
BE1	Mailing personnalisé	Les administrateurs peuvent envoyer des mails à chaque groupe ou adhérent à partir d'un formulaire	Email envoyé et redirection vers le tableau de bord avec un message d'erreur ou de succès	OK
BE2	Ajout des avatars (Upload)	Les administrateurs ajoutent les avatars de chaque membre à partir de la liste des adhérents	Ajoute l'image dans un dossier de l'arborescence de fichiers de l'application et intègre une nouvelle URL dans la BDD pour l'attribut lié à l'avatar	OK

FIGURE 4.11 – Tableau des plans de tests (Extrait)

4.1 Modifications des objectifs

Voir les annexes => Compte-rendu de réunion 11/01/16

Après une seconde réunion avec la cliente Mme Isabelle Léonard, plusieurs questions se sont posées entraînant alors des modifications dans le cahier des charges.

1. Dans le cas des maquettes fonctionnelles dévoilées précédemment, est-il envisageable qu'un utilisateur puisse avoir plusieurs associations ?
2. Peut-il y avoir plusieurs administrateurs dans une association et peut-on changer le rôle des membres dans la gestion des adhérents ?
3. Comment différencier les anniversaires des activités de manière visuelle ?
4. Pour la modification des avatars de chaque membre, est-il possible d'exporter un trombinoscope ?

Afin de répondre à ces questions, il est nécessaire de réfléchir à des solutions techniques et fonctionnelles envisageables dans les délais fixés.

Dans le cas de la première question, il est nécessaire de modifier à la fois l'interface de connexion et la composition de la base de données. La technologie côté serveur utilisée pour ce projet gère facilement des méthodes d'authentification (Voir Firebase => Partie Technique) et il est possible en retour d'une inscription d'ajouter un utilisateur au sein de la base de données. En d'autres termes, on gardera en mémoire l'authentification de l'utilisateur à l'aide de son adresse mail et on ajoutera un élément dans l'association lui correspondant. En effectuant l'analogie entre ces deux éléments, il sera possible à l'utilisateur d'avoir plusieurs associations. Par ailleurs, il faudra ajouter à la page de connexion un champ correspondant à une liste des associations. Ainsi, l'utilisateur n'aura qu'à inscrire son identifiant et son mot de passe dès qu'il aura en amont choisi l'association dans laquelle il veut entrer.

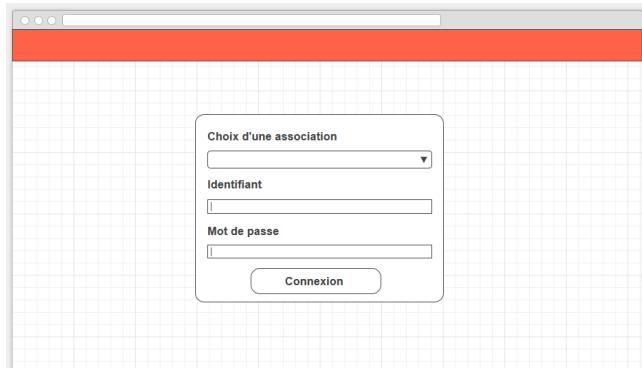


FIGURE 4.12 – Nouvelle page de connexion (Maquette fonctionnelle)

Lors de l'inscription de l'association en base, il était décidé d'inscrire les informations personnelles de l'administrateur. Ainsi, son compte se créait en même temps que l'association. Il était donc automatiquement lié à l'association. La disparition de son compte occasionnait celle de l'association. Ce système manque de flexibilité et il a été réfléchi d'apporter une nouvelle interface à la page d'inscription et une nouvelle disposition dans la base de données pour les administrateurs.

Cette fois, seules la deuxième et troisième étapes de la page d'inscription ont été gardées. En d'autres termes, il est possible d'ajouter autant de membres dans l'association que l'on souhaite et de lui attribuer un nom et une description (modifiable et affichée sur le tableau de bord). Lors de l'ajout des membres, il est possible de modifier le rôle de chacun : administrateur ou simple adhérent. L'affichage différera selon le rang de l'utilisateur.

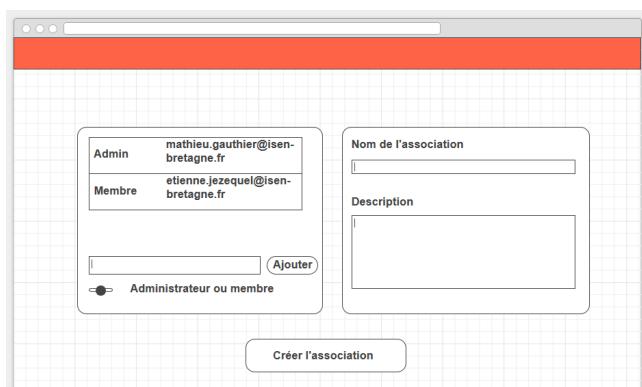


FIGURE 4.13 – Nouvelle page d'inscription (Maquette fonctionnelle)

Ensuite, la cliente Mme Isabelle Léonard précisait que la différenciation entre les anniversaires et les activités est indispensable pour améliorer l'ergonomie de l'application. Sachant que les utilisateurs

peuvent avoir des niveaux relativement faibles en informatique, il est important de mettre en valeur tel ou tel événement de manière à ne pas les induire en erreur.

Pour cela, il est possible d'attribuer à chaque événement un type ("birthday" ou "events") dans la base de données. Selon ce type, l'apparence visuel de l'événement sera différente. En vert, seront affichés les anniversaires et en bleu, les activités.

Les technologies utilisées pour l'exportation de fichiers en PDF sont relativement récentes et ne prennent pas forcément en compte toutes les fonctionnalités nécessaires à la génération d'un trombinoscope. La meilleure manière d'intégrer des images au sein d'un tableau regroupant les noms, adresses, prénoms, numéros de téléphone de chacun des membres contenus dans une association est d'intégrer un chemin de données traduit en Base 64. Toutefois, une image d'une taille supérieure à 150 pixels de largeur se traduit par une grande quantité d'informations en Base 64. Dans ce cas, il sera nécessaire de limiter la taille des fichiers lors de l'importation de ces images.

5 Méthodologie

Quelle méthodologie utiliser pour ce type de projet ?

La méthodologie utilisée pour la gestion de ce projet est basée sur une méthodologie agile.

Les méthodes agiles se base sur trois piliers :

- **La transparence** : ces méthodes utilisent un langage commun. Ce qui permet à tout observateur d'avoir une compréhension rapide du projet
- **L'inspection** : cela permet de faire des points régulièrement sur différents aspects du produit afin d'éviter de s'éloigner de l'objectif principal du projet.
- **L'adaptation** : dans le cas où l'équipe s'est éloignée du but du projet, le travail à fournir est alors ré-adapté grâce à des *rituels*. Ces rituels sont, par exemple, la réunion de planification de sprint, la mêlée quotidienne, la revue de sprint ainsi que la rétrospective de sprint.

En effet, chacun des membres a un rôle défini (chef de projet, équipe de développement, celle de management...). Cependant avec une équipe de deux personnes, il est difficile de vraiment représenter tous les rôles. Il était donc nécessaire d'endosser plusieurs rôles.

De plus, des stand-up étaient organisés tous les matins entre développeurs. C'est une réunion quotidienne de quelques minutes où les membres de l'équipe se retrouvent pour discuter de ce qu'ils ont fait la veille et de ce qu'ils vont faire dans la journée. Lors de cette réunion, ils exposent leurs problèmes (si présents) et chacun peut aider ou conseiller les autres membres. En effet, un oeil extérieur peut parfois trouver des solutions à un problème récurrent.

En complément des stand-up, des réunions ont été mises en place afin d'effectuer des états d'avancement avec le client. Ces réunions peuvent s'apparenter à des revues de sprint. Un sprint est le nom d'une itération. Cette itération dure 30 jours calendaires en théorie, mais en pratique entre 2 et 4 semaines. Chaque sprint a un but. Pendant une itération, l'équipe doit développer une liste de fonctionnalités du produit qui a été définie au début du sprint. Une revue de sprint sert justement à valider ces fonctionnalités développées pendant le sprint.

Le système de sprint ne fut pas employé dans ce projet. En effet, le sprint n'est pas vraiment adapté pour ce type de réalisation car il n'y avait pas de période régulière de projet fixé. Mais les revues de sprint de la méthode agile ont beaucoup inspirées la méthodologie mise en place lors de ce projet.

Les tâches à réaliser pour le projet étaient divisées en deux catégories :

- **Back-End** : la partie non visible par l'utilisateur (le coeur de l'application)
- **Front-End** : la partie visible et utilisée par l'internaute (l'interface utilisateur)

Ainsi, en catégorisant ces tâches, en leur définissant une priorité et une durée judicieusement définie d'un commun accord entre membres du groupe, les problèmes furent moins présents et mieux appréhendés. Chaque développeur réalise et teste de manière autonome les tâches à réaliser en fonction de leur priorité. Après chaque réunion d'avancement et validation des nouvelles fonctionnalités (développées et testées), un déploiement est effectué. Grâce à cette méthode il existe toujours une version stable et utilisable du produit.

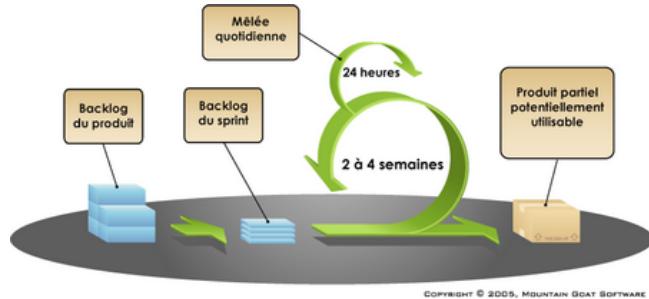


FIGURE 4.14 – Schéma de la méthode agile

6 Outils de développement et de gestion

Dans le but d'informer le client de l'avancement du projet, il fut mis en place certains outils. Il ne s'agit pas de tous les logiciels ou dispositifs mis en place. En effet, certains sont utilisés pour la technique et l'environnement de travail. Ces derniers seront détaillés dans une prochaine section liée aux choix technologiques.

6.1 Wrike

Wrike, tout comme Trello, est un outil en ligne (SaaS) pour la gestion de projet et la collaboration de travail. Il permet de gérer l'état d'avancement du projet et les attributions des tâches à chaque membre de l'équipe. Les délais et les horaires de développement sont pris en charge et permettent plus facilement de lier la méthodologie agile au logiciel Wrike. Ainsi, les développeurs peuvent avec facilité ordonner et administrer un tableau d'avancement des tâches.

De plus, les documents relatifs à la gestion de projet sont regroupés sur le cloud Wrike permettant à la fois, de ne pas perdre des informations mais aussi de les transmettre à toute l'équipe. Des notifications envoyées par mail ou par l'intermédiaire de l'application mobile permettent de rester au courant des mises à jour du tableau et de réagir par rapport à celles-ci.

6.2 Tableau d'avancement

Une demande a été effectuée au début du projet pour installer un tableau physique d'avancement des tâches dans le bureau de la cliente Mme Isabelle Léonard. L'objectif est de collaborer non seulement entre développeurs mais aussi entre client et réalisateurs.

La communication au sein d'un projet est indispensable pour rester fidèle au but principal et au cahier des charges. Ainsi, cela permet de ne pas se disperser sur des tâches moins essentielles. De plus, les réunions régulières sont de bonnes occasions pour mettre à jour le tableau et montrer l'avancement du projet à la cliente.

Le tableau se résume à quatre colonnes distinctes :

- **Une colonne "à faire"** contenant toutes les tâches indispensables à la finalisation du projet.

- "**En cours**" : il s'agit des tâches en développement qui n'ont pas encore été testées
- "**En test**" : il s'agit des tâches qui ne sont plus en développement mais qui n'ont pas encore totalement été finalisées. Cela demande de multiples tests : failles de sécurité, problèmes graphiques, expérience utilisateur simplifiée...
- "**Validée(s)**" : il s'agit des tâches complètement validées et sur lesquelles il n'y a plus besoin de revenir. Ces tâches peuvent alors être mises en production.

6.3 GitHub

Git est un logiciel de gestion de versions décentralisé. Il est conçu pour être efficace avec n'importe quel projet. Contrairement à des outils comme SVN³, Git fonctionne de façon décentralisée, c'est-à-dire que le développement ne se fait pas sur un serveur centralisé, mais chaque personne peut développer sur son propre dépôt sur son poste de travail en local. Git facilite ensuite la fusion (merge) des différents dépôts.

Un logiciel de gestion de versions est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées. Il permet par exemple de retrouver facilement les différentes versions d'un lot de fichiers et de savoir qui a pu modifier tel fichier à l'origine d'un problème. Il est donc facile de s'organiser et d'avoir un suivi très rigoureux qui permet un développement efficace.

La gestion du répertoire Git s'est faite avec GitHub. Il s'agit d'une application web. Depuis cette interface web, il est possible de suivre les modifications sur le répertoire avec la présence d'un file d'actualité ou "timeline".

Cette outil est indispensable car il permet de bien gérer les versions du produits. En cas de problème, il est possible de revenir en arrière à une version stable du produit. De plus, tout est sauvegardé sur un dépôt distant, le code source n'est donc pas perdu.

3. SVN : logiciel de gestion de versions, distribué sous licence Apache et BSD

Chapitre 5

Partie technique

La partie technique est l'une des phases les plus importantes du projet. En effet, les technologies utilisées tout au long de celui-ci sont très récentes et permettent de nettement améliorer les performances techniques de l'application Web. Le choix en matière de technologies a surtout été porté vers la partie cliente de manière à minimiser le nombre de lignes de code du côté serveur.

1 Choix technologiques

1.1 Partie cliente

Actuellement, AngularJS est une technologie prédominante dans le développement d'application Web. Il permet d'ajouter une approche "Full-Stack"¹ à une application avec une architecture MVC présentant de très bonnes idées.

Cependant, le développement d'une telle application n'est pas toujours simple. En effet, l'API des "directives" de Angular n'est pas évidente à prendre en main et peu intuitif. Voici un schéma explicatif du fonctionnement de AngularJS (pour plus d'informations, voir le glossaire) :

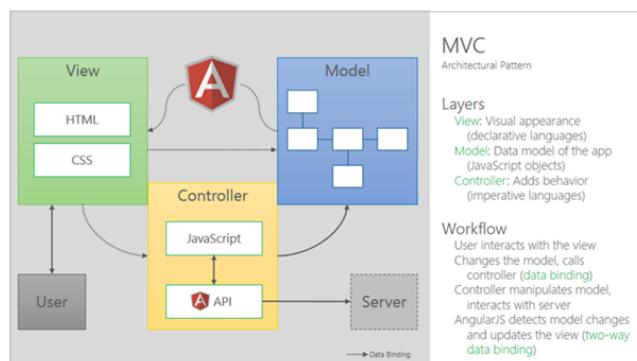


FIGURE 5.1 – Schéma explicatif du fonctionnement de Angular

C'est pour cette raison qu'une nouvelle technologie du nom de ReactJS développée par Facebook ne mise pas sur le côté framework Full-Stack. Par ailleurs, React n'est pas du tout un framework. Il ne s'agit seulement que d'une bibliothèque dite orientée Composant.

1. Full-Stack : capable de gérer à la fois le Front-End et le Back-End

Qu'est ce qu'un composant ?

Un composant selon ReactJS contient des propriétés d'initialisation (appelées Props), des états (appelés States), une méthode de rendu (render) et des fonctions liées au cycle de vie du composant (componentWillMount, componentDidMount...).

L'application Web comporte un ou plusieurs composants et chacun de ses composants sont chargés et rechargés à chaque changement d'un état. Les méthodes de cycle de vie permettent alors d'agrémer l'application d'événement tel qu'une entrée en base ou modification de l'interface graphique. C'est ici qu'intervient le **Virtual-DOM**.

Le Virtual DOM est une technique utilisant une panoplie de librairies/d'algorithme qui permet d'améliorer considérablement les performances du Front-End en évitant de travailler directement avec le DOM mais plutôt uniquement avec un objet Javascript très léger qui imite un arbre du DOM. **Mais à quoi sert véritablement le virtual DOM ?**

Tout d'abord, il est indispensable de comprendre la définition du DOM. Le DOM (ou real DOM) est une API (interface de programmation d'application, par exemple Jquery est une API utilisant du Javascript) permettant de définir la structure logique d'un document HTML et d'un XML. On représente un tel document sous la forme d'un arbre. On parle d'un arbre du DOM.

Par exemple : React crée un arbre d'objets (un ensemble de composants) représentant une partie du DOM. Il s'agit du virtual DOM. Au lieu de créer un élément réel <DIV> contenant un , React va créer un objet React.div qui contient un object React.ul.

React peut alors manipuler ces composants très rapidement sans pour autant travailler dans le DOM réel. Quand il doit afficher un composant, il utilise le virtual DOM pour déterminer de quoi il a besoin pour faire correspondre les deux arbres DOM : celui créé par le DOM et celui créé par React. Sachant que les mises à jour des données se font essentiellement sur un arbre virtuel du DOM, l'application Web gagne en performance et en rapidité.

Immutabilité et le flux à sens unique

Le concept d'immutabilité (un objet ne change pas, on en crée un nouveau à chaque changement) est un point que React a bien compris. Cela permet de garder en mémoire des états de composant et de les réutiliser quand cela est nécessaire. On n'écrase jamais l'information d'un état, on en crée un nouveau en passant par une variable tierce.

Contrairement à AngularJS, les données de l'application sont envoyées par un flux unidirectionnel par l'intermédiaire des propriétés initiales (Props) et des états (state) des composants. En d'autres termes, dans React, on définit à un composant parent des états et si on veut les transférer vers le composant enfant, on l'effectue par le biais des Props (propriétés initiales). Cela produit un effet de chaîne qui permet à la fois de structurer l'application de manière logique et d'éviter les problèmes d'héritage entre composants. Il n'est pas possible de transférer des états depuis l'enfant vers le parent.

Une autre manière de coder des Web Apps : le JSX

Depuis des années, la séparation des responsabilités entre la vue déclarative en HTML et la logique impérative en Javascript est incontournable. AngularJS respecte ce principe. React, quant à lui, prend le contre-pied et propose une manière de coder en déclaratif directement dans le retour de la méthode de rendu (render). Il s'agit du langage JSX. Par ailleurs, cela nécessite une transpilation² en Javascript pour être correctement compris par les moteurs d'exécution JS des navigateurs.

Et les données ?

Cependant, l'utilisation de React, qui ne permet que de créer des composants injectables dans le DOM, pose des questions quant à l'architecture logicielle qu'il faut utiliser : comment aller chercher la donnée ? Comment mettre à jour les composants lorsque la donnée change ? On pourrait fusionner React et Angular qui possède justement une approche MVC classique. Toutefois, la philosophie de la technologie de Facebook n'est pas vraiment respectée par Angular. L'alternative efficace se nomme Redux.

Redux n'est autre qu'une variante améliorée de Flux, un système d'architecture mis en place par Facebook pour éviter les soucis liés à des applications complexes comprenant une centaine de composants et nécessitant des liens simples entre ces derniers. Dans Redux, quatre termes sont indispensables à comprendre :

- **Action** : Si on veut modifier l'affichage d'un composant ou déclencher un comportement, on est obligé d'utiliser une action dans une architecture MVC type Redux. Avec une action, on va pouvoir modifier un état (state) d'un composant React. Par exemple, une action *INCREMENT* peut augmenter la valeur de 1 d'un état et une autre action *DECREMENT* peut diminuer sa valeur de 1.
- **Dispatcher** : il s'agit du composant unique qui reçoit toutes les actions de l'application. En fonction d'un comportement d'un composant React, le Dispatcher va notifier les stores que l'action a eu lieu. Par exemple, un simple clic sur un bouton va déclencher l'action *INCREMENT*.
- **Store** : il s'agit d'un composant de Redux qui va contenir et gérer les états de l'application. Il sert d'intermédiaire entre l'architecture Redux et l'application Web.
- **Reducer** : Une action va décrire le fait que quelque chose s'est effectuée à tel moment mais elle ne va pas spécifier comment l'état de l'application change en réponse à cette action. C'est le rôle du Reducer. Par exemple, dans le cas de l'action *INCREMENT*, c'est le Reducer qui va modifier la valeur de l'état de l'application. Si l'état initial est de 0, avec l'action *INCREMENT*, le Reducer le fera passer à 1.

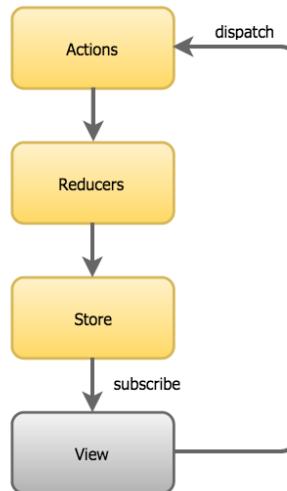


FIGURE 5.2 – Schéma explicatif du fonctionnement de Redux

Comment gérer la partie design et graphique ?

Comme expliqué dans la présentation du sujet, la cliente recherche un outil ergonomique et intuitif, utilisable par quelqu'un possédant peu de compétences informatiques. L'objectif est clair. On veut du

2. Transpilation : compilation d'un code source Javascript en un code HTML/CSS/JS adapté aux navigateurs actuels

"clé en main".

L'apport du design dans cette application Web ne doit pas se résumer à une simple tendance actuelle. Il doit permettre de synthétiser les principes d'une bonne conception d'interface.

La première version fonctionnelle du produit présentée lors d'une réunion d'état d'avancement avec la cliente utilisait Bootstrap, une librairie CSS et Javascript (Voir Glossaire) utilisant le système de grilles (permet d'adapter l'application à des tailles de résolutions plus faibles comme les mobiles ou les tablettes). Cependant, son design est quelconque et peu parlant. Il n'arrive pas à mettre en avant les éléments visuels indispensables à la navigation.

Après un stand-up consécutif à la réunion, il était important de combiner à la fois une technologie capable de réaliser les prouesses de Bootstrap en matière de Responsive Design et un concept graphique pouvant améliorer considérablement l'expérience utilisateur. Par ailleurs, cette combinaison ne doit en aucun cas déclencher des conflits au sein du code. La première technologie doit donc être extrêmement minimalistes et miser seulement sur le système de grille et l'autre devait seulement proposer des atouts visuels.

Voici les deux choix technologiques en matière de Design :

- **SimpleGrid** est une librairie de style CSS. Elle reprend le même concept de grille que bootstrap avec la partie responsive. Mais contrairement à ce dernier, SimpleGrid est dépourvue de style de mise en forme. La librairie est très légère et permet d'avoir une interface à la fois très lisible sur mobile et optimisée sur les plus grandes résolutions.
- **Material-UI** est une librairie de style CSS et de Javascript. Sa particularité réside dans le fait qu'elle est adaptée essentiellement à des applications Web développées avec ReactJS. Elle permet de définir facilement des composants utilisant des éléments visuels du Material Design. Ce dernier est un ensemble de règles de mise en page et de mise en forme d'interface web. Ce design a été proposé par Google. Il est utilisé dans android et la majorité des applications de google.

Le choix du Material Design n'est pas anodin. En effet, il est déjà généralisé sur les applications mobiles. Leurs utilisateurs ont déjà été confrontés à ce type de Design. Il offre une grande interactivité par la présence de nombreuses animations fluides mais impose aussi une hiérarchie dans la page en donnant des impressions de perspectives (travail sur les ombres). Cela permet d'améliorer la navigation au sein du produit et offre alors une meilleure utilisabilité contrairement à Bootstrap.

1.2 Partie serveur

Comme expliqué précédemment, il était important de ne pas s'occuper complètement de la gestion des données au sein même du projet. Il est souvent compliqué d'éviter les failles de sécurité, les dégradations de performances et de maîtriser les systèmes d'authentification.

Une solution SaaS : Front-End Sans Back

Firebase est une solution SaaS qui prend en charge toute la partie serveur permettant alors aux développeurs de créer des applications web uniquement du côté client. Firebase s'appuie sur une API complète proposant une gestion complète de la sécurité et de l'authentification.



FIGURE 5.3 – Logo Firebase

L'un des atouts majeurs de la solution est l'extensibilité de la base de données Firebase permettant de supporter un grand nombre d'utilisateurs connectés sans aucune baisse de performances (Scalabilité). L'autre avantage important est la synchronisation automatique des données entre les clients et le serveur. En d'autres termes, cela permet de réaliser facilement un système de messagerie en temps réel (par exemple). Si un client ajoute ou modifie une donnée, tous les autres clients observent cette modification de leur côté instantanément.

La partie relativement compliquée à mettre en place pour une équipe de développeurs est de prévoir les failles de sécurité. L'avantage de Firebase est sa suite complète de méthodes d'authentification permettant aux développeurs d'avoir un contrôle total sur la connexion d'un utilisateur. Pour cela, il suffit de mettre en place des règles de sécurité très simples et flexibles.

De plus, le trafic est sécurisé de manière à assurer que les données ne soient jamais lues ou modifiées par un tiers.

Voici un aperçu de la BDD générée par Firebase fonctionnant avec des objets et tableaux JSON.



FIGURE 5.4 – Schéma JSON généré par Firebase

Firebase n'est pas comme toutes les autres bases de données. Elle est complètement "scalable" et gérée en temps réel. De plus, toutes les données sont stockées dans une structure en arbre.

Firebase peut seulement supporter les transactions sur les sous-arbres de cette struture et elle assure l'intégrité des données par l'intermédiaire d'un langage de sécurité. Il est, en effet, possible de créer une liste de restrictions en JSON.

1.3 Pourquoi de tels choix technologiques ?

React et Firebase est une excellente combinaison en matière de technologies. Mais pourquoi prendre en compte ces choix alors qu'il en existe d'autres tout aussi efficaces ?

L'efficacité est une chose. Mais ce n'est pas, à première vue, l'objectif fixé. Le but principal de ce projet est de miser sur la simplicité. Le concept très fonctionnel de React agrémenté de l'absence de prise en charge du côté serveur par Firebase permet de rendre le projet modulable à souhait par une autre équipe de développeurs.

Les documentations de React et de Firebase sont très claires et précises. Le code peut paraître plus verbeux mais plus on progresse avec React et Firebase plus il devient absolument logique et cohérent. La courbe d'apprentissage est très rapide, en une semaine on a fait vite le tour et touché aux fonctionnalités avancées. L'énorme avantage est que lorsqu'on vient d'écrire un composant même simple, on a exploré 70% de la librairie.

2 Environnement de travail

Node JS

Node.js est un serveur open-source en JavaScript orientée vers les applications réseau qui doivent pouvoir gérer une montée en charge, soit sa capacité à maintenir ses fonctionnalités et ses performances en cas de forte demande.

Node.js contient une bibliothèque de serveur HTTP intégrée, ce qui rend possible de faire tourner un serveur web sans avoir besoin d'un logiciel externe comme Apache. Il permet de faire du JavaScript en dehors du navigateur pour ajouter de nouvelles fonctionnalités à l'application. Il propose une nouvelle façon de développer des applications web dynamiques.

Node.js n'est pas un framework. Node.js est un environnement bas niveau. Il se rapproche donc plus du C que de PHP.

Node a la particularité d'être rapide grâce à deux choses : le moteur V8 et son fonctionnement non bloquant. En effet ce moteur javascript, (utilisé dans google chrome par exemple) exécute du code javascript très rapidement. De plus, comme JavaScript est un langage conçu autour de la notion d'évènement, Node.js a pu mettre en place une architecture de code entièrement non bloquante. Une architecture non bloquante consiste à exécuter le code de manière asynchrone (Le processus appelant continue de travailler pendant que le processus appelé exécute le traitement demandé et gère via un événement le retour du processus appelé).

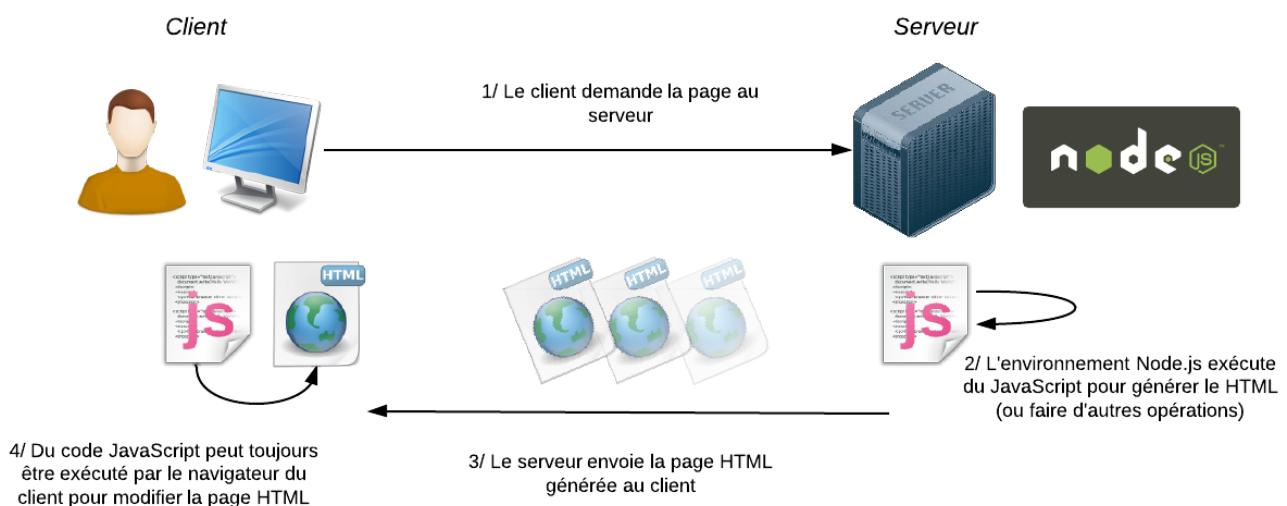


FIGURE 5.5 – Schéma explicatif du fonctionnement de Node JS

Webpack

Webpack est une solution très utilisée dans les environnements de développement pour sa capacité à faciliter l'organisation d'une application en modules.

En d'autres termes, Webpack dispose d'un système permettant de créer un pack prêt pour développer ou déployer sur le Web. D'un point de vue plus technique, il s'agit de l'équivalent de Maven³ en Java mais pour du Javascript. Les dépendances en matière de modules pour l'application Web s'effectuent dans un fichier Package.json. Celui-ci regroupe les informations relatives à l'application ainsi que les commandes permettant de transpiler, de déployer et de tester le projet.

3. Maven : Moteur de production dont la fonction principale est d'automatiser la (re)construction de l'application

Heroku

Heroku est une plate-forme permettant d'effectuer du Cloud Computing c'est-à-dire de pouvoir héberger de manière simple les applications Web. Il permet de faire le lien entre le dépôt Git de l'application et le serveur distant de herokuapp.com. Par ailleurs, cette connexion peut automatiser en continu le déploiement de l'application. A chaque fois que le dépôt Git bénéficie d'une modification, le serveur effectue une mise à jour. De plus, Heroku s'applique à de nombreux langages informatiques : Ruby, Rack, Java, Python, JS...

Pourquoi l'utiliser ?

Afin de prévenir régulièrement Mme Isabelle Léonard d'une mise à jour du produit, il a été mis en place un déploiement via Firebase. Toutefois, ce dernier ne permet qu'une mise en production statique. En d'autres termes, toute la partie côté serveur n'est pas prise en compte. Dès l'implémentation de l'ajout des avatars et du service de mailing, il était indispensable d'avoir une prise en charge complète de la partie serveur. Firebase ne pouvant plus gérer cette fonctionnalité, il était essentiel de rechercher une nouvelle solution capable de supporter Node JS et Webpack. Heroku est l'un des seuls produits à proposer gratuitement ce type de service.

3 Exploitation des résultats

3.1 Résultat de l'application finale

(a) Page d'accueil de l'application

(b) Tableau de bord d'un administrateur

(c) Edition d'un membre par un administrateur

FIGURE 5.6 – Résultat final de l'application Web

3.2 Futures améliorations envisageables

Le développement de l'application a permis d'envisager de futures améliorations. Tout est mis en oeuvre pour qu'une nouvelle équipe de développeurs puisse reprendre le projet et intégrer de nouvelles fonctionnalités. Voici une liste des ajouts potentiels à intégrer dans le futur :

- **Nouvelle version de Firebase :** Pour la première fois depuis le rachat de Firebase, Google propose lors de la Google I/O (du 18 mai 2016) une nouvelle version largement remaniée proposant de nouvelles fonctionnalités dont la mise en place de notifications sur mobile, un système de stockage de fichiers, des tests unitaires, des envois de mails d'inscription, la gestion des rapports de bug...

La refonte de l'application Web avec cette nouvelle version permettrait d'améliorer nettement les performances en éliminant petit à petit la partie serveur propulsée par Node JS. Dans ce cas, seule la partie cliente sera à gérer. Tous les traitements en base de données seront administrés par le BackEnd-As-A-Service de Firebase.

- **Des thèmes plus personnalisés** : Material-UI comporte un système de palette de couleurs très complet qui permettrait de styliser son tableau de bord comme l'administrateur le souhaiterait. Dans l'application originelle, seuls cinq thèmes sont créés au préalable dont celui de l'ISEN. Il s'agit d'un choix de développement afin d'éviter d'avoir à effectuer un grand nombre de scénarios d'usage, lors de la phase de tests.
- **Porter l'application à un stade au dessus de la gestion d'associations** : Le développement de l'application permet de diversifier son activité à un autre stade. Il est possible de porter l'application à une nouvelle spécificité comme par un petit réseau social ou un outil de communication au sein d'une entreprise.
- **Application mobile (IOS, Android)** : Il sera possible grâce à Firebase et React Native (Version de développement de React pour les applications natives mobiles) de reprendre les grandes lignes de l'application avec l'ajout de fonctionnalités comme le partage de photos, la gestion de la localisation GPS. Par ailleurs, grâce à la nouvelle version de Firebase, on peut mettre en place un système de notifications de manière simple.

Chapitre 6

Conclusion

Ce projet M1 est la conclusion de la première année de Master à l'ISEN Brest. En imposant une gestion de projet stricte et une méthodologie suivie, le projet a été rendu en adéquation avec les charges initiales et dans les délais définis.

De plus, les choix technologiques effectués pendant ce projet permettent une exportation du produit à des fins plus généralisées tel qu'un réseau social ou un système de communication instantanée. Ainsi, le développement permet la réutilisabilité du code par une autre équipe de développeurs.

En Août 2016, la solution sera utilisée auprès d'un véritable public avec l'association de la cliente Mme Isabelle Léonard. Les retours négatifs comme positifs de cette première expérience seront les bienvenus afin d'assurer la maintenance et l'évolution du produit.

D'un point de vue technique, malgré la jeunesse des technologies utilisées, elles commencent à prendre de l'ampleur et parviennent à se créer des écosystèmes très puissants. Ces derniers permettront alors de faire évoluer ces solutions dans le bon sens.

Chapitre 7

Annexes

Comptes-rendus

Compte Rendu de Réunion							
Date	15/12/15	Objet	Première réunion : application native ou web	Date début	15/12 - 17h30		
Lieu	ISEN BREST			Date fin	15/12 - 18h30		
Participants	Isabelle Léonard Mathieu Gauthier Etienne Jézéquel	Firmes/ Services	M1	Rédacteur	Etienne Jézéquel		
				Diffusion			
Compte rendu			Actions	Responsable	Délais		
Quel type d'application ? - native ou Web			Choix du Web : client plus léger	Mathieu Gauthier Etienne Jézéquel			
Compréhension et définition des exigences fonctionnelles, il y a t il des exigences techniques?			pas d'exigences techniques. exigences fonctionnelles à rédiger dans sous forme de cahier des charges.	Mathieu Gauthier Etienne Jézéquel			
Mise en place d'un tableau d'avancement et explication de la méthodologie utilisée pendant le projet.			Demande d'un tableau d'avancement.	Mathieu Gauthier Etienne Jézéquel			

FIGURE 7.1 – Compte-rendu de réunion du 15/12/2015

Compte Rendu de Réunion							
Date	05/01/16	Objet	Élaboration du cahier des charges	Date début	05/01 - 17h30		
Lieu	ISEN BREST			Date fin	05/01 - 18h30		
Participants	Isabelle Léonard Mathieu Gauthier Etienne Jézéquel	Firmes/ Services	M1	Rédacteur	Etienne Jézéquel		
				Diffusion			
Compte rendu			Actions	Responsable	Délais		
Quelles sont les couleurs de l'interface web ainsi que la couleur dominante? - Charte graphique			FE2	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Avez vous des éléments visuels à intégrer dans l'interface (ex: photo d'adhérents) ? - Intéressé par les photos des adhérents - système de galerie intéressant			FE2	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Un logo ?			FE2	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Avec vous des idées sur la Police d'écriture? - Charte graphique ISEN			FE2	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Avez vous des idées de design ? - Nous avons montré une première maquette. Elle a plus. Nous partirons sur ce modèle. Mockup à réaliser.			FE2	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Croquis de maquette.			Prototype	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Présentation d'exemple de design de site.			FE2	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Avez vous des idées de fonctionnalités supplémentaires ? - fil d'actualité -> oui - que se passe - t - il lors de la suppression d'un grand nombre d'adhérents au sein d'un groupe - espace utilisateur et administrateur sécurisé -> Sécurité maximum, génération d'un mot de passe par défaut envoyé aux utilisateurs inscrits par le directeur de l'association lors de l'inscription. - planning présence / absence (temps réel) -> Oui			BE6	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Ordre de la page principale ?			FE1	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Arborescence du site, grandes sections ?			FE1	Mathieu Gauthier Etienne Jézéquel	Jeudi 7 Janvier		
Gantt trop détaillé. Refaire un planning général du projet et un planning de développement détaillé.			Revoir le planning prévisionnel	Mathieu Gauthier Etienne Jézéquel	Vendredi 8 Janvier		

FIGURE 7.2 – Compte-rendu de réunion du 05/01/2016

Compte Rendu de Réunion							
Date	11/01/15	Objet	Réunion d'état d'avancement	Date début	11/01 - 17h00		
Lieu	ISEN BREST			Date fin	11/01 - 18h00		
Participants	Isabelle Léonard Mathieu Gauthier Etienne Jézéquel	Firmes/ Services	M1	Rédacteur	Etienne Jézéquel		
				Diffusion			
Compte rendu			Actions	Responsable	Délais		
Présentation du mockup : présentation d'une maquette fonctionnelle afin de présenter les fonctionnalités du projet ainsi que les navigations entre les différentes pages.			Modification du cahier des charges à modifier, par conséquent la maquette également.	Mathieu Gauthier Etienne Jézéquel			
complément des exigences du client :			Notes : <ul style="list-style-type: none"> - On doit pouvoir avoir plusieurs admin et on peut changer de roles. - On peut différencier les activités. - Trombinoscope ? - Un utilisateur peut avoir plusieurs associations. 				
Tableau d'avancement			Mise en place des différentes colonnes d'avancement et des post-it. Modification de l'avancement.	Mathieu Gauthier Etienne Jézéquel			

FIGURE 7.3 – Compte-rendu de réunion du 11/01/2016

Bibliographie

- [1] Call-Em-All. Material ui documentation. <<http://www.material-ui.com/>>, 2015.
- [2] Facebook Developer. React tutoriel. <<https://facebook.github.io/react/>>, 2014.
- [3] Mozilla Developer. Mozilla. <<http://developer.mozilla.org/>>, 2016.
- [4] inc Github. Github. <<http://github.com/>>, 2016.
- [5] Google and Firebase Team. Firebase. <<http://www.firebaseio.com/>>, 2016.
- [6] Stack Overflow. Stack overflow forum js. <<http://stackoverflow.com/>>, 2016.
- [7] Heroku Team. Heroku documentation. <<http://heroku.com/>>, 2016.
- [8] NodeJS Team. Nodejs. <<http://nodejs.org/>>, 2016.