

Dynamic_Assoc'

Évaluation des outils – Version 1.4

Historique des révisions

Date	Version	Description	Auteur(s)
12.11.2017	1.0	Rédaction initiale	AF
13.11.2017	1.1	Mise en page et orthographe	SG
04.12.2017	1.2	Corrections suite à A2, ajout des explications des matrices, ajout de MS Project dans les outils utilisés	AF, SG
Avril 2018	1.3	Modifications suite à l'arrêt de la collaboration avec le CODAP	SG
Mai 2018	1.4	Correction du choix du framework et mise en page finale	SG

Table des matières

I.	1.Choix des outils de développement.....	2
	1.1 Quelle solution d'hébergement utiliser ?.....	2
	Comparaison entre les différents types d'hébergement.....	2
	Matrice préférentielle	4
	Matrice multicritère	5
	1.2 Quel logiciel de développement utiliser ?	6
	Matrice préférentielle	6
	Matrice multicritères	6
	1.3 Quel logiciel de gestion des versions utiliser ?	7
	Matrice préférentielle	7
	Matrice multicritères	8
	1.4 Autres logiciels utilisés ?	8
	1.5 Faut-il utiliser un Framework ? Si oui, lequel ?.....	9
	Matrice préférentielle	10
	Matrice multicritères	10
	Quel Framework utiliser ?.....	11
II.	2. Annexe	12
	2.1 Explication des matrices préférentielles	12

1. Choix des outils de développement

Avant de se lancer dans le développement de notre solution et de débiter la phase de construction, il est nécessaire de déterminer quels outils utiliser pour mener à bien cette phase.

1.1 Quelle solution d'hébergement utiliser ?

Le choix de l'hébergement ne porte pas sur quel fournisseur choisir, mais quel type d'hébergement choisir. Le tableau ci-dessous va nous aider à choisir entre deux différents types d'hébergements proposés chez Infomaniak.


Comparaison entre les différents types d'hébergement

Caractéristiques	Hébergement Web	Serveur Cloud Managé
Capacité d'espace disque	<ul style="list-style-type: none">- 100 Go d'espace disque SSD (extensible jusqu'à 500 Go)- Ceci concerne uniquement les fichiers. La quantité d'informations présentes dans les bases de données n'est pas comptée dans cet espace disque. Ceci est donc amplement suffisant pour le stockage des vidéos, documents et autres fichiers que les associations souhaitent mettre à disposition.- Hébergement partagé sur serveur Dell DSS 1500 Family- CPU Intel Xeon 8 cœurs- 96 Go RAM- 1280 Mo de mémoire RAM pour l'exécution des scripts	<ul style="list-style-type: none">- 100 Go d'espace disque SSD (extensible jusqu'à 1 To)- 2 CPU (extensible jusqu'à 24 CPU)- 6 Go RAM (extensible jusqu'à 50 Go)- 2048 Mo de mémoire RAM pour l'exécution des scripts
Base de données utilisée	<ul style="list-style-type: none">- Base de données MySQL illimitées- Utilisation de PHP/MyAdmin en dernière version- Les bases étant illimitées, aucun problème concernant le nombre de comptes utilisateurs pouvant être créés	<ul style="list-style-type: none">- Base de données MySQL illimitées- Utilisation de PHPMyAdmin en dernière version- Supporte également MongoDB
Sauvegarde	<ul style="list-style-type: none">- Sauvegarde quotidienne de l'ensemble des données (fichiers, bases de données, comptes FTP, etc)- Les données étant sauvegardées quotidiennement, il n'y a aucun souci à se faire concernant une éventuelle perte de données	<ul style="list-style-type: none">- Sauvegarde quotidienne de l'ensemble des données (fichiers, bases de données, comptes FTP, etc)

Caractéristiques	Hébergement Web	Serveur Cloud Managé
Sécurité des données	<ul style="list-style-type: none"> - Certificats SSL gratuits - Un certificat SSL sécurise les données que les utilisateurs saisissent. - Permet une connexion en https:\\ - Infomaniak fournit également une protection face aux attaques de type DDoS 	<ul style="list-style-type: none"> - Certificats SSL gratuits - Infomaniak fournit également une protection face aux attaques de type DDoS
Stockage des données	<ul style="list-style-type: none"> - Les données sont stockées en Suisse - Ceci évite les problèmes de fuite de données dans d'autres pays 	<ul style="list-style-type: none"> - Les données sont stockées en Suisse
Trafic	<ul style="list-style-type: none"> - Trafic illimité, aucune limite de trafic mensuel 	<ul style="list-style-type: none"> - Trafic illimité, aucune limite de trafic mensuel
Disponibilité	<ul style="list-style-type: none"> - Infomaniak s'engage à offrir une disponibilité de ses services à 99,99% Ceci fait que l'accès au site web ainsi qu'à ses services sera disponible en tout temps par les utilisateurs. 	<ul style="list-style-type: none"> - Infomaniak s'engage à offrir une disponibilité de ses services à 99,99%
Type de langage	<ul style="list-style-type: none"> - PHP avec possibilité de choix de la version (7 / 5.6 / 5.5) 	<ul style="list-style-type: none"> - PHP avec possibilité de choix de la version (7 / 5.6 / 5.5) - Supporte également Git, Node.js, Python, Pearl...
Type de connectivité	<ul style="list-style-type: none"> - Connectivité ultra-rapide avec plus de 60 Gbps de bande passante - Ceci fait que les utilisateurs n'auront pas de problèmes de latence et délais de chargement des pages du site internet. 	<ul style="list-style-type: none"> - Connectivité ultra-rapide avec plus de 60 Gbps de bande passante
Hébergement	<ul style="list-style-type: none"> - Jusqu'à 20 sites web 	<ul style="list-style-type: none"> - Jusqu'à 100 hébergements et 400 sites web

Matrice préférentielle¹

Afin d'évaluer quel type d'hébergement serait le plus adéquat pour notre situation, nous avons réalisé une matrice préférentielle :

No.		a	b	c	d	e	f	g	h	i
	Éléments de solution									
a	Capacité d'espace disque									
b	Base de données illimitées	b								
c	Sauvegarde des données	c	c							
d	Sécurité des données	a	b	c						
e	Trafic	a	b	c	d					
f	Uptime	a	b	c	d	f				
g	Type de langage	a	b	c	d	e	f			
h	Type de connectivité	a	b	c	d	h	h	h		
i	Hébergement	a	b	c	d	e	f	g	h	

No.	Éléments de solution	Nombre (36)	Ordre	Pondération %	Pondération modérée %
a	Capacité d'espace disque	6	3	16,66	18,00
b	Base de données illimitées	7	2	19,44	18,00
c	Sauvegarde des données	8	1	22,22	22,00
d	Sécurité des données	5	4	13,88	14,00
e	Trafic	2	7	5,55	6,00
f	Uptime	3	6	8,33	8,00
g	Type de langage	1	8	2,77	2,00
h	Type de connectivité	4	5	11,11	11,00
i	Hébergement	0	9	0,00	1,00

Nous comparons chaque numéro les uns avec les autres afin de déterminer lesquels sont les plus importants dans notre situation.

Grâce à cela, nous pouvons déterminer la pondération d'un point et ensuite réaliser une matrice multicritères qui nous indiquera quel type d'hébergement est le plus adapté à nos besoins.

^{1 1} Une explication des matrices préférentielle est donnée en dernière page du document (page 12)

Matrice multicritère

Grâce à ces matrices nous pouvons évaluer les différentes solutions selon ces critères, avec les pondérations définies précédemment.

Cette évaluation passe par une analyse dite multicritères :

		Hébergement Web		Serveur Cloud Managé	
Critères	Pondération	Note	Note pondérée	Note	Note pondérée
Capacité d'espace disque	6	8	48	10	60
Base de données illimitées	7	8	56	10	70
Sauvegarde des données	8	10	80	10	80
Sécurité des données	5	9	45	9	45
Trafic	2	10	20	10	20
Uptime	3	10	30	10	30
Type de langage	1	5	5	10	10
Type de connectivité	4	10	40	10	40
Hébergement	1	5	5	10	10
Σ			225		235

Après analyse nous nous apercevons, comme pensé initialement, que la différence entre les deux solutions est minime.

Connaissant les coûts de chacune des deux solutions, nous pouvons réaliser l'intégration du coût pour celles-ci :

	Hébergement Web	Serveur Cloud Managé
Coût total (sur 1 année)	128,52 CHF	505,44 CHF
Total des points (valeur de la solution)	225	235
Coût par point	0,57 CHF	2,15 CHF

Bien que la différence entre les deux types d'hébergement soit minime, l'intégration du coût est par contre bien différente. La solution pour le serveur managé revient presque quatre fois plus coûteuse que l'hébergement web classique.

Nous avons donc décidé, d'opter pour un hébergement web classique qui, de par ses fonctionnalités, correspondra très bien au développement prévu.

De plus, il arrive que les associations soient à but non lucratif, donc ayant tenu en compte ce détail, il a été décidé d'opter pour un hébergement peu onéreux.


1.2 Quel logiciel de développement utiliser ?

Il existe de nombreux environnements de développement intégrés, mais afin de pouvoir faire un choix parmi ceux-ci nous avons également réalisé une matrice de préférence.

Pour cela, nous avons sélectionné trois logiciels qui sont :

- NetBeans
- Microsoft Visual Studio
- Adobe DreamWeaver

Matrice préférentielle

No.		a	b	c	d	e	f	g
	Éléments de solution							
a	Connaissance de l'environnement							
b	Compatibilité avec les équipements	b						
c	Environnement en français	a	b					
d	Documentation de qualité	d	b	d				
e	Facilité d'utilisation	e	b	e	e			
f	Synchronisation FTP Infomaniak	f	f	f	f	f		
g	Support du langage PHP	g	b	g	g	g	f	

No.	Éléments de solution	Nombre (21)	Ordre	Pondération %	Pondération modérée %
a	Connaissance de l'environnement	1	6	4,77	6,00
b	Compatibilité avec les équipements	5	2	23,85	20,00
c	Environnement en français	0	7	0,00	3,00
d	Documentation de qualité	1	5	4,77	5,00
e	Facilité d'utilisation	3	4	14,31	16,00
f	Synchronisation FTP Infomaniak	6	1	28,62	30,00
g	Support du langage PHP	4	3	19,08	20,00

Nous avons donné beaucoup d'importance au point concernant la synchronisation avec le compte FTP d'Infomaniak, car grâce à cela nous pourrions travailler directement sur les fichiers composant le site internet ainsi que la base de données et les synchroniser directement depuis l'environnement de développement. Nous pourrions ainsi visualiser directement les changements effectués.

Matrice multicritères

		NetBeans		Microsoft Visual Studio		Adobe DreamWeaver	
Critères	Pondération	Note	Note pondérée	Note	Note pondérée	Note	Note pondérée
Connaissance de l'environnement	1	8	8	5	5	3	3
Compatibilité avec les équipements	5	10	50	10	50	10	50
Environnement en français	1	8	8	10	10	10	10
Documentation de qualité	1	7	7	10	10	8	8
Facilité d'utilisation	3	7	21	8	24	7	21
Synchronisation FTP Infomaniak	6	1	6	8	48	10	60
Support du langage PHP	4	10	40	10	40	10	40
Σ			82		132		139

Après analyse, nous nous apercevons d'une quasi-équivalence entre les environnements de développement que sont Visual Studio et DreamWeaver.

Étant étudiants à la HEG, le logiciel de Microsoft est gratuit alors que celui d'Adobe, bien que faisant des prix étudiants, a quand même un coût d'acquisition.

C'est pour cette unique raison que nous allons utiliser Visual Studio pour le développement de notre projet.

1.3 Quel logiciel de gestion des versions utiliser ?

Afin de pouvoir travailler efficacement en équipe, nous allons également avoir besoin d'un logiciel de gestion des versions.

Ce type de logiciel permet de créer une arborescence de fichiers afin de conserver toutes les versions de ces derniers, ainsi que leurs différences (gestion différentielle). C'est un outil indispensable lorsque l'on souhaite développer de manière collaborative, car il permet au groupe de développeurs de stocker toutes les évolutions du code source pour chaque développeur et conserve un historique.

Cela facilite également les maintenances ultérieures, chose importante dans la solution que nous allons développer.

Il existe plusieurs systèmes de versioning, mais nous en avons retenu deux, proposés par défaut dans Microsoft Visual Studio.

Il s'agit de :

- Visual Studio Team Service
- GitHub

Matrice préférentielle

No.		a	b	c	d	e
	Éléments de solution	→				
a	<i>Plugin pour Visual Studio</i>					
b	<i>Documentation sur le plugin</i>	a				
c	<i>Documentation sur la solution</i>	c	c			
d	<i>Outils agiles inclus</i>	a	b	c		
e	<i>Prise en main facile</i>	e	e	c	e	

No.	Éléments de solution	Nombre (10)	Ordre	Pondération %	Pondération modérée %
a	<i>Plugin pour Visual Studio</i>	2	3	20,00	18,00
b	<i>Documentation sur le plugin</i>	1	4	10,00	8,00
c	<i>Documentation sur la solution</i>	4	1	40,00	35,00
d	<i>Outils agiles inclus</i>	0	5	0,00	4,00
e	<i>Prise en main facile</i>	3	2	30,00	35,00

Nous pouvons maintenant évaluer ces deux solutions au travers d'une matrice multicritères.

Matrice multicritères

		Team Service		GitHub	
Critères	Pondération	Note	Note pondérée	Note	Note pondérée
Plugin pour Visual Studio	2	10	20	10	20
Documentation sur le plugin	1	10	10	8	8
Documentation sur la solution	4	8	32	8	32
Outils agiles inclus	1	10	10	0	0
Prise en main facile	3	7	21	8	24
Σ			63		56

Suite à cette analyse, nous avons opté pour l'utilisation de l'outil Visual Studio Team Service pour le développement de notre projet. Ceci est un plus, mais il est à noter que cet outil à l'avantage d'offrir de nombreux autres outils qui s'ajoutent à ses fonctions principales de versioning.

Ces outils, pensés pour le développement agile, DAD dans notre cas, permettent de gérer intégralement la Work Item List.

- Définir les Work Items avec les éléments suivants :
 - Points
 - Priorité
 - Risque
- Assigner les Work Items aux membres de l'équipe
- Planifier les itérations
- Évaluer la charge de travail de chaque membre
- Tableau de bord intégré présentant les Work Items de l'itération en cours
- Burndown Chart

Il y a un certain nombre d'autres outils, mais non nécessaire dans notre cas. Nous n'en ferons donc pas allusion.

1.4 Autres logiciels utilisés ?

En plus des logiciels cités précédemment, nous avons également décidé d'utiliser l'outil de planification MS Project. Les raisons qui nous ont poussé vers cet outil sont simples : l'outil est disponible gratuitement pour les étudiants et nous avons déjà utilisé cet outil, nous le connaissons bien donc le choix nous a paru évident. De plus, nous n'avons pas trouvé utile de faire des comparaisons avec d'autres outils puisque nous l'utilisons uniquement pour la planification Gantt et pour réaliser le diagramme des Use-Cases. Pour faire ces deux documents MS Project suffit amplement.

1.5 Faut-il utiliser un Framework ? Si oui, lequel ?

Un framework permet de réaliser de manière structurée des sites complexes et cela avec un code clair et maintenable.

Le terme framework signifie en français "cadre de travail". Ce cadre est un ensemble de composants servant à créer les fondations, l'architecture et les grandes lignes d'un logiciel.

Un framework est une boîte à outils conçue par un ou plusieurs développeurs à destination d'autres développeurs. C'est donc un outil permettant d'améliorer la productivité des développeurs et d'avoir une base de travail où il n'est pas nécessaire de repartir à zéro à chaque nouveau projet.

Cependant, lorsque l'on souhaite utiliser un nouveau produit, il est nécessaire de peser le pour et le contre.

Avantages d'un framework :

- Gain de productivité
- Incite, de par sa propre architecture, à réaliser un bon code et un code bien structuré est un code facilement maintenable et évolutif
- Offre des « briques » prêtes à être utilisées, ce qui évite de réinventer la roue
- Ces briques sont développées par des équipes chevronnées, elles sont donc flexibles et robustes
- Structure et conventions de code connues
- Communautés entourant chaque framework (tutoriels, cours, forums)
- Documentation de qualité et régulièrement mise à jour
- Support à long terme garanti et mises à jour qui ne cassent pas la compatibilité

Inconvénients d'un framework :

- Courbe d'apprentissage élevée
 - Avant de pouvoir l'utiliser de manière optimale, il est nécessaire d'apprendre comment il fonctionne.
- Architecture imposée
 - Le projet devra être réalisé en fonction de celle-ci, nous avons donc une certaine perte de liberté
- Dépendance
 - Un projet développé avec un framework doit le rester par la suite. Le changement de framework est souvent lourd et coûteux.
- Lourdeur
 - Certains framework peuvent être relativement lourds par rapport au projet en lui-même.

Comme on peut le constater, il y a beaucoup d'avantages à utiliser un framework, mais les inconvénients sont tout de même non négligeables.

C'est pour cette raison que nous avons réalisé une matrice préférentielle afin d'évaluer l'intérêt ou non d'utiliser un framework.

Matrice préférentielle

No.		a	b	c	d	e	f	g
	Eléments de solution							
a	Connaissance de l'environnement							
b	Connaissance du langage de programmation	b						
c	Durée de développement	c	b					
d	Compatibilité avec les équipements	d	d	d				
e	Déploiement multiplateforme	a	b	c	d			
f	Liberté d'architecture	a	b	c	d	f		
g	Dépendance avec la plateforme	g	b	c	d	g	g	

No.	Eléments de solution	Nombre (21)	Ordre	Pondération %	Pondération modérée %
a	Connaissance de l'environnement	2	5	9,54	10,00
b	Connaissance du langage de programmation	5	2	23,85	22,00
c	Durée de développement	4	3	19,08	18,00
d	Compatibilité avec les équipements	6	1	28,62	25,00
e	Déploiement multiplateforme	0	7	0,00	5,00
f	Liberté d'architecture	1	6	4,77	5,00
g	Dépendance avec la plateforme	3	4	14,31	15,00

Nous allons maintenant évaluer ces choix par le biais d'une matrice multicritères.

Matrice multicritères

		Framework		Développement web "natif"	
Critères	Pondération	Note	Note pondérée	Note	Note pondérée
Connaissance de l'environnement	2	0	0	7	14
Connaissance du langage de programmation	5	7	35	7	35
Durée de développement	4	8	32	6	24
Compatibilité avec les équipements	6	10	60	10	60
Déploiement multiplateforme	1	10	10	0	0
Liberté d'architecture	1	0	0	8	8
Dépendance avec la plateforme	3	7	21	0	0
Σ			123		92

Suite à ces résultats et malgré un temps d'apprentissage certain, nous avons décidé d'utiliser un framework. Cela nous confèrera une bonne base départ et surtout nous permettra d'utiliser un code structuré et maintenable.

De plus, tout bon développeur ne travaille pas sans framework de nos jours.

Quel Framework utiliser ?

Il existe trois types de framework permettant le développement de sites internet.

- Framework basés sur des langages web
- Framework basés sur des langages orientés software (Java)
- Framework basés sur leurs propres langages de programmation

Notre choix va se porter sur le premier type de framework. Une des raisons principales est que tous les membres de notre équipe ont déjà des connaissances dans plusieurs langages, que ça soit HTML, JavaScript ou encore PHP. De plus, notre hébergement chez Infomaniak nous oblige à développer en PHP. Une fois ce premier choix effectué, reste à savoir quel framework utiliser parmi ceux du premier type abordé.

Suite à de nombreux échanges avec Infomaniak, ils nous ont vivement conseillé d'opter pour le framework Bootstrap, c'est pour cette raison que nous n'allons pas effectuer d'étude approfondie sur le choix d'un framework en particulier.

« Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. » (*Wikipédia*)²

Bootstrap contient des codes HTML et CSS qui vont grandement nous aider pour la mise en forme et le design de notre solution. De plus, il y a énormément de développeurs de modules Bootstrap que nous pouvons utiliser gratuitement sans avoir à les développer nous-même. Cela nous aide grandement puisque nous n'avons pas de connaissances en design d'interfaces.

Avant l'arrivée de Bootstrap, il y avait plusieurs bibliothèques qui étaient utilisées, cependant, le fait d'utiliser plusieurs bibliothèques différentes menait à des incohérences et rendait la maintenance plus difficile. Ayant opté pour la méthodologie DAD, nous but est d'offrir une maintenance aisée et utiliser Bootstrap nous permet d'aller dans ce sens.

Deux sites internet (www.openclassroom.com et www.developpez.net) abordent très largement ce framework en proposant de nombreuses aides. Nous allons donc pouvoir démarrer notre apprentissage sur celui-ci sereinement et pourrons également compter sur une communauté active.

² [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))

2. Annexe

2.1 Explication des matrices préférentielles

La matrice préférentielle contient des éléments de solution que nous jugeons important à prendre en compte dans le présent projet. Selon le projet, ces éléments peuvent différer.

Chose importante, aucun élément d'aspect financier ne doit apparaître dans cette matrice.

Si besoin est, le coût est étudié dans "le coût par point" afin de donner plus de finesse à l'analyse.

Pour la lecture et compréhension :

Chaque élément de solution est comparé, un à un, avec les autres éléments.

À chaque comparaison, de manière subjective, est défini quel élément nous semble le plus important/pertinent. Ce dernier est indiqué dans la case correspondante (cellule de croisement).

Une fois toutes les comparaisons effectuées :

- On indique le nombre total de comparaisons
- Le nombre fois que chaque élément apparaît (nombre de fois où l'élément est vainqueur d'une comparaison)
- Un ordre est défini en fonction du nombre d'apparitions
- Une pondération est effectuée $((100/\text{nbTotal}) * \text{nbApparitions})$
- Celle-ci peut être légèrement adaptée (pondération modérée) si le résultat nous semble trop petit ou trop grand. Mais il faut que cette modification, comme son nom l'indique, soit modérée, car sinon la matrice ne sert au final à rien.

Suite à cela, les résultats sont reportés dans la matrice dite multicritère.

Les valeurs de l'onglet « Pondération » correspondent au nombre de fois que l'élément est apparu dans la matrice préférentielle.

L'onglet « Note », contient également des valeurs subjectives, basées sur notre expérience, connaissance ou sur des recherches, comparatifs effectués, trouvés.

L'onglet « Note Pondérée » est simplement le résultat de la Pondération * Note

L'addition de ces notes pondérées nous donne un score global de chaque solution comparée.

Si l'aspect financier est important, alors il est possible, comme indiqué, de réaliser une intégration de coût (1 an, 2 an, n années). Il suffit de diviser le coût total de la solution par le nombre total de points obtenus par la solution afin d'avoir un coût par point.

En effet, une solution peut avoir un score global beaucoup plus élevé certes, mais elle peut également avoir un coût par point très élevé, ce qui la rend peut-être moins intéressante aux yeux du mandant.

Le choix sera fait par le mandant, nous ne lui apportons que des éléments l'aidant dans la prise de décision de ce choix.