



République Algérienne Démocratique Populaire

Université Saad Dahlab Blida-1

Faculté de Technologie



Département des Énergies Renouvelables
Master 2 conversion photovoltaïque

Mémoire du
Projet Personnel en Énergies Renouvelable

Thème :
**Conception et développement d'une
application software pour le dimensionnement
des installations photovoltaïques**

Réaliser Par Bouraib Zakia
 Mokhtar Asma

Soutenu devant le jury composé par :

DR. Rachid Boukenoui	USDB	Président
DR. AitSahed Oussama	USDB	Examineur
DR. Maazouz Halima	USDB	Examineur
DR. Amrouche Badia	USDB	Encadreur

Septembre 2022

الملخص

يعد إنتاج الطاقة تحديا ذا أهمية كبيرة للسنوات القادمة في الواقع تتزايد باستمرار احتياجات الطاقة في المجتمعات الصناعية علاوة على ذلك ستحتاج البلدان النامية الى المزيد والمزيد من الطاقة لتنفيذ تنميتها بشكل جيد لذا فان الطاقات المتجددة هي الحل الامثل لتلبية هذه الحاجة , كيفية الاستفادة المثلى من هذه الطاقة دفعت بالتفكير نحو الموضوع الحالي الذي هو موضوع دراستنا , وهو: تصميم وتطوير تطبيق سطح مكتب لأبعاد النظام كهروضوئي الذي سيسهل عمل المصممين من خلال مراعات البيانات الجغرافية لموقع التثبيت واحتياجات الطاقة والمعدات المتوفرة في السوق , حتى يتمكن المصمم من اختيار عدة حلول واختيار الحل الذي يضمن توازنا مرضيا بين المواصفات الفنية و الميزانية المتاحة.

الكلمات المفتاحية : التركيب الكهروضوئي , تطبيق سطح مكتب

Résumé

La production d'énergie est un défi de grande importance pour les années à venir, en effet, les besoins énergétique des sociétés industrialisées ne cessent augmenter par ailleurs, les pays en voie de développement auront besoin de plus en plus d'énergie pour mener à bien leur développement. Donc l'énergie renouvelable constitue des solutions pour répondre à ce besoin. L'optimisation de l'utilisation de cette énergie est à l'origine de l'objet de notre étude , à savoir: conception et développement d'une application software de dimensionnement de systèmes photovoltaïques qui facilitera le travail des concepteurs en prenant en considération les données géographiques de site d'installation , les besoins énergétique et les équipements disponibles sur le marché ainsi , le concepteur pourra tester plusieurs solutions et choisir celle qui assure un équilibre satisfaisant entre les spécifications techniques et budget disponible.

Mots clés : dimensionnement, système photovoltaïque, application software.

Abstract

The energy is a challenge of great importance for the coming years. Indeed, the energy needs of industrialized societies are constantly increasing. Countries will need more and more energy to carry out their development. Renewable energy is a solution to optimize the use of this energy. This pushed the reflection towards the present theme, which is namely: design and development of a software application for PV systems dimensioning. This software will facilitate the designers' tasks by taking into account the geographical data of the site, the energy needs and the equipment available on the market, so the designer can test several solutions and choose the one that ensures a satisfactory balance between the technical specifications and the available budget.

Keywords: photovoltaic systems, sizing, software application.

Dédicaces

Nous dédions ce modeste travail particulièrement à ceux qui nous ont apprises le sens de la vie, nous ont encouragées et nous ont soutenues durant nos études.

- *Nos très chers parents.*
- *A nos frères.*
- *A tous les membres de nos familles*

BOURAIB et MOKHTAR

Nous tenons à remercier tous nos amis qui nous ont aidées de loin ou de près dans la réalisation de ce modeste travail.

Remerciements

*Je remercie tout d'abord ALLAH pour m'avoir donnée la volonté,
la force, le courage
Et surtout la santé*

*Mes profonds remerciements vont à mes parents de m'avoir
soutenue moralement et financièrement durant ces longues années.*

*Ensuite je remercie mes sœurs et frère Chahrazed, Farida et Rabeh,
pour leur soutien tout au long de mes années scolaires.*

*Plus particulièrement :
Notre promoteur AMROUCHE BADIA de nous avoir orientées
significativement tout au long de ce travail.*

*Je remercie le monsieur Mohamad Lounnas, qui eut le plus grand
crédit pour la création de cette application et pour m'avoir
accompagnée jusqu'au dernier moment*

*Je remercie aussi docteur Boukhalfa Mohammed pour l'orientation
qu'il m'a apportée*

Sans oublier mon amie Amina Zamiche et tous mes amis

*Merci aussi à nos professeurs à qui
Nous exprimons tout notre respect
et profonde gratitude*

BOURAIB-ZAKIA

Remerciements

J'adresse mes remerciements à toutes les personnes qui m'ont aidée dans la réalisation de ce mémoire.

En premier lieu, je tiens, à exprimer ma gratitude et mes remerciements à Monsieur le Directeur De SKTM et à Monsieur le Directeur du centre de recherche de Ghardaïa UREAR pour avoir accepté notre stage et pour m'avoir donnée la chance de me fother une petite expérience dans ces lieux d'excellence en énergie renouvelables.

*je veux remercier : mes sœurs Aya et Bassma Et mon frère Issam
Et mes amies Khaoula, Chafia, Rania, Iman
ET Anwar Yahiaoui
Et tous mes amis*

MOKHTAR-ASMA

Nomenclature

Symbole	Désignation	Désignation
pv	Photovoltaïque	Photovoltaïque
E_c	Energie consommée	Energy consumed
P	Puissance	power
I_r	Irradiation	Irradiation
Pu	Puissance utile	Power user
CdTe	Cadmium Telluride	Cadmium Telluride
AsGa	Arsenium de gallium	Gallium arsenium
CIS	Cuivre indium sélénium	Copper indium sélénium
EPV	Energie photovoltaïque	Solar energy
GAP	groupe de physique appliqué	Applied physics group
HOMER	Hybride Optimisation de Multiple Energie Ressources	Hybrid optimization of multiple energy resources
PC	Computer personnel	Personal computer
ASCII	Code standard américain pour l'échange d'informations	American standard code for information interchange
DE	Département de l'énergie	Energy department
NREL	National Renouvelable Energie Laboratoire	National renewable energy laboratory
SAM	SYSTEM ADVISOR MODEL	SYSTEM ADVISOR MODEL
CEC	Californie Electricité Commission	California new and renewable éléctricity
MNRE	Ministre des énergies nouvelles et renouvelables	Ministre of new & renewable energy
CAO	Conception assistée par ordinateur	Computer aided design
UML	Modélisation oriente objet	Unified modeling langage
SPA	Application mono page	single page application
HTML	Langage de balisage hypertexte	Hyper text markup langage
css	Feuilles de style en cascade	Cascading style sheets
http	Protocole de transfert de texte hyper	Huper text transfer protocol
UC	Cas d'utilisateur	use case
UI	l'interface d'utilisateur	User interface
RVB	Ruge verre Bleu	Red green blue
ppp	Point par pause	Point per step
API	Interface de programmation d'application	Application programing interface
VS Code	Visuel studio code	Visuel studio code
soda	Solar radiation data	Solar radiation data
GPV	Générateur photovoltaïque	Photovoltaic générateur
Wh	Watt heure	Watt hour
w	watt	watt
v	volt	volt
Ah	Ampère heure	Ampère hour
WC	Watt crête	Watt crête

Sommaire

RÉSUMER	I
Dédicace	II
Remercîment	III
Symbole	IV
Sommaire	V
Liste des figures	VI
Liste des tableaux	VII
Introduction générale	1
Chapitre 1 : L'état de l'Art sur les logiciels de dimensionnement des installations photovoltaïque	
1.1 Introduction	1
1.2 Les étapes de la vie d'un projet photovoltaïque	2
1.3 Problématique	3
1.4 Étude de l'existant	4
1.4.1 Homer PRO	5
1.4.2 PV Planner	6
1.4.3 PV F-CHART	7
1.4.4 RET SCREEN	8
1.4.5 PV système	9
1.4.6 SYSTEM ADVISOR MODEL (SAM)	10
1.4.7 Solar pro	10
1.5 Comparaison entre pvsyst et Homer pro	10
1.6 Cahier de charge	11
1.7 Conclusion	11
Chapitre 2 : Conception de l'application software	
2.1 Introduction	12
2.2 Application web monopage	12
2.3 Présentation de l'application	13
2.3.1 Description des fonctionnalités de l'application	13
2.3.2 Diagramme de communication entre différentes interfaces :	13
2.3.2.1 Diagramme dynamique	14
2.3.2.2 Diagramme structure (statistique)	16
2.3.3. Les données nécessaires de l'application	18
2.4. Structure de l'interface d'utilisateur (UI)	19
2.4.1 La charte graphique de l'application	19
2.4.2 Maquettage avec adobe Illustrator	20
2.5.3 programme d'arrière-plan de l'application	24
2.5 Présentation de l'application	34
2.6 Conclusion	39
Chapitre 3 : Développement de l'application software	
3.1 Introduction	40
3.2 Description du programme (Vue.js) :	40
3.2.1 programme des équations de dimensionnement	40
3.2.2 Irradiation annuelle :	41
3.2.3 Énergie produit	45

3.3 Transformation de programme par électrons	46
3.3.1 Description de la méthode de transformation	46
3.3.2 Présentation de résultat de transformation	47
3.4 Conclusion	50
Chapitre 4 : Présentation d'une cas d'étude	
4.1 Introduction	51
4.2 Dimensionnement par notre logiciel	51
4.3 Dimensionnement par logiciel PVSYST	55
4.4 Comparaison entre les deux résultats	58
4.5 Conclusion	58
Conclusion générale	59
Références	60

List des figures

Figure 1.1: logiciel Homer pro version 3.12	4
Figure 1.2 : logiciel PV planner	5
Figure 1.3 : logiciel PV F-chart	6
Figure 1.4: logiciel PV RETscreen	7
Figure 1.5 : logiciel pvsyst	9
Figure 1.6 : logiciel Solar pro	10
Figure 1.7: système advisor	10
Figure 2.1 : Historique d'UML	14
Figure 2.2 : Schéma complet du processus de modélisation d'une application web.	14
Figure 2.3 : schéma de diagramme de cas d'utilisateur par logiciel ArgoUML (réalisation personnelle)	15
Figure 2.4 : schéma de diagramme de séquence par logiciel ArgoUML (réalisation personnelle)	16
Figure 2.5 : schéma de diagramme de classe par ArgoUML (réalisation personnelle)	17
Figure 2.6: schéma de diagramme d'activité par ArgoUML (réalisation personnelle)	18
Figure 2.7 : l'idée générale de réalisation de logo	19
Figure 2.8 : logo de l'application software	20
Figure 2.9 : capture de nouveau document dans Adobe Illustrator	20
Figure 2.10 : capture de choix de couleur dans Adobe Illustrator	21
Figure 2.11 : capture de 1 ^{ème} phase de l'application par Illustrator	21
Figure 2.12 : capture de menu latéral	22
Figure 2.13 : capture de 2 ^{ème} phase de l'application par Illustrator	22
Figure 2.14 : capture de 3 ^{ème} phase de l'application par Illustrator	23
Figure 2.15 : capture de 4 ^{ème} phase de l'application par Illustrator	23
Figure 2.16 : capture de 5 ^{ème} phase de l'application par Illustrator	24
Figure 2.17: capture de fichier bootstrap	24
Figure 2.18 : capture 1 de code css	24
Figure 2.19 : capture 2 de code css	23
Figure 2.20 : capture 3 de code css	25
Figure 2.21: capture 4 de code css	26
Figure 2.22 : capture 5 de code css	26
Figure 2.23 : capture 6 de code css	27
Figure 2.24 : capture 1 de code html pour bar d'option	27
Figure 2.25 : capture 2 de code html pour bar d'option	28
Figure 2.26: capture 3 de code html pour bar d'option	28
Figure 2.27 : capture 1 de code html pour le menu latéral	29
Figure 2.28 : capture 1 de code JavaScript pour le menu latéral	29
Figure 2.29: capture 1 de code html pour section1	29
Figure 2.30: capture 2 de code html pour section1	30
Figure 2.31: capture 3 de code html pour section1	30
Figure 2.32: capture 4 de code html pour section1	30
Figure 2.33: capture 1 de code html pour section2	30
Figure 2.34: capture 2 de code html pour section2	31
Figure 2.35: capture 1de code html pour section 3	31
Figure 2.36: capture 2 de code html pour section 3	31
Figure 2.37: capture 3 de code html pour section 3	32
Figure 2.38: capture 4 de code html pour section 3	32
Figure 2.39: capture 5 de code html pour section 3	32
Figure 2.40: capture 6 de code html pour section 3	33

Figure 2.41: capture 1 de code html pour section 4	33
Figure 2.42: capture 1 de code html pour section 5	33
Figure 2.43: capture 1 de code vue.js pour les déférant section	34
Figure 2.44 : capture 2 de code vue.js pour les déférant section	34
Figure 2.45: capture 1 de code de déférant bouton	34
Figure 2.46 : capture de 1 ^{ème} phase de l'application	35
Figure 2.47 : capture de 1 ^{ème} phase exprimé le fonctionnement du bouton (suivant)	35
Figure 2.48 :capture de 1 ^{ème} phase exprimé le fonctionnement du bouton (fichier)	36
Figure 2.49 : capture 1 ^{ème} phase exprimé le fonctionnement du bouton (aide)	36
Figure 2.50 : capture de 1 phase exprimer le menu latéral	36
Figure 2.51 : capteur de 2 ^{ème} phase de l'application	37
Figure 2.52: capture de 3 ^{ème} phase de l'application	37
Figure 2.53 : capture de 4 ^{ème} phase d'application	38
Figure 2.54 : capture de 5 ^{ème} phase de l'application	38
Figure 3.1 : déclaration des variables de table de besoin et choix des équipements	41
Figure 3.2: fonction de vue.js pour calculer les besoin et choix des équipements	41
Figure 3.3 : phase 2 choix de site géographique	42
Figure 3.4 : caractéristique vue.js pour choix de site géographique	42
Figure 3.5 : fonction vue.js pour choix de site géographique	42
Figure 3.6 : interface de site web (Solar radiation data « soda »)	43
Figure 3.7 : capture 1 fichier Excel d'irradiation	43
Figure 3.8 : capture 2 fichier Excel d'irradiation	43
Figure 3.9 : table html pour les données de l'irradiation	44
Figure 3.10: fonction Vue.js pour l'affichage	44
Figure 3.11: résultat 1 de La fonction dataIrr ()	44
Figure 3.12: résultat 2 de La fonction dataIrr ()	45
Figure 3.13: programme de calcule l'énergie produit d'un système pv	45
Figure 3.14: programme d'affichage de graphe l'énergie produit d'un système pv	46
Figure 3.15: téléchargement node et npm	46
Figure 3.16: création de fichier électronjs	46
Figure 3.17: téléchargement des fichiers git dans le fichier électronjs	47
Figure 3.18 : capture Transformation de programme par électronjs	47
Figure 3.19 : les fichiers de notre application software	48
Figure 3.20 : phase 1 de notre application software	48
Figure 3.21 : phase 2de notre application software	48
Figure 3.22 : phase 3 de notre application software	49
Figure 3.23 : phase 4 de notre application software	49
Figure 3.24 : phase 5 de notre application software	49
Figure 4.1 : bouton de notre application(Dimpower)	51
Figure 4.2 : 1 ^{er} interface de notre application(Dimpower)	52
Figure 4.3 : choix de site géographique et autre caractéristique par notre application	52
Figure 4.4 : table de l'irradiation	53
Figure 4.5 : table de besoin par notre application (Dimpower)	53
Figure 4.6 : les choix des équipements de système par notre application (Dimpower)	54
Figure 4.7 : le résultat final de notre simulation par notre application(Dimpower)	54
Figure 4.8 : l'énergie produit de système photovoltaïque	55
Figure 4.9 : les cordonnée géographique de site dan logiciel Pvsyst	55
Figure 4.10 : les données de l'éclairement du site de Blida dans Pvsyst	56

Figure 4.11 : table de besoin par logiciel Pvsyst	56
Figure 4.12 : le choix de type de batterie par Pvsyst	56
Figure 4.13 : le choix de type de panneau par Pvsyst	57
Figure 4.14 : l'énergie produit de système photovoltaïque par Pvsyst	57
Figure 4.15 : la résultat final de notre simulation par Pvsyst	58

Liste des tableaux

Tableau 1.1 : Comparaison entre pvsyst et Homer pro :	10
Tableau 1.2 : récapitulatif des contraintes générales	11
Tableau 4.1 : résumé de résultat de simulation par notre application software	55
Tableau 4.2 : résumé de résultat de simulation par logiciel pvsyst	58

Introduction générale

Aujourd'hui, les énergies renouvelables deviennent progressivement des énergies à part entière, rivalisant avec des énergies fossiles du point de vue coût et performance de production [1, 2]. De ces énergies nouvelles (biomasse, éolienne, solaire, géothermie), le solaire photovoltaïque présente une convenance particulière pour notre pays, puisqu'il dispose d'un gisement solaire fortement propice au développement de cette forme d'énergie.

Les résultats des travaux relatifs au dimensionnement des systèmes photovoltaïques tels que PVSYST, PVSOL et Homer pro sont largement utilisés car ils constituent des outils d'aide à la décision sur les énergies renouvelables et font partie des outils d'aide au dimensionnement les plus utilisés [3]. Ces logiciels constituent pour la plupart, des outils d'aide à la décision dans des projets sur des systèmes photovoltaïques. Lorsque ceux-ci effectuent un calcul de générateur photovoltaïque, ils se limitent juste à un pré dimensionnement du système étudié. De plus, Ils sont complètement adaptés au contexte européen et n'intègrent pas de données qui rendraient leur utilisation profitable aux pays en voie de développement.

Dans la grande majorité, ces pays ne disposent pas d'un outil d'aide au dimensionnement des systèmes photovoltaïques propre à eux et à leur contexte et intégrant les données solaires qui leurs sont propres. Un outil, qui en plus de rendre la tâche facile aux ingénieurs dans des projets relatifs au photovoltaïque, constituerait un excellent moyen de vulgarisation du photovoltaïque.

La mise sur pied d'une plateforme à partir des outils informatiques simples apportera aux ingénieurs et techniciens des pays en développement, beaucoup d'aisance et de facilité dans le dimensionnement des systèmes photovoltaïques domestiques. La production d'électricité photovoltaïque sera facilitée à travers une obtention rapide du coût estimatif du générateur photovoltaïque calculé pour une puissance installée donnée [4-5].

Pour cet objectif, dans ce travail, nous avons développé une application software pour le dimensionnement d'un système photovoltaïque à partir du langage de développement web. Pour ce faire, le mémoire est organisé en quatre chapitres et une conclusion générale.

Le premier chapitre décrit l'Etat de l'art sur les logiciels de dimensionnement des installations photovoltaïques. La problématique est y exposée également. Le but principal de ce chapitre est de présenter une synthèse comparative entre deux logiciels (PVsyst et Homer pro) et le cahier de charge de notre application.

Dans le deuxième chapitre, nous présenterons les interfaces de cette application, les organigrammes de communication entre différentes interfaces.

Dans le troisième chapitre un accent est mis sur le côté programmation tandis que le quatrième chapitre présentera les résultats d'un exemple d'étude.

Finalement, une conclusion générale donne une synthèse du projet.

Chapitre 1 :

L'état de l'Art sur les logiciels de dimensionnement des installations photovoltaïque

1.1 Introduction

Energie solaire photovoltaïque désigne l'énergie récupérée et transformée directement en électricité à partir de la lumière du soleil par des panneaux photovoltaïques. Elle résulte de la conversion directe dans un semi-conducteur (silicium, le CdTe, l'AsGa, le CIS, etc.) d'un photon en électron. Outre les avantages liés à l'absence de maintenance des systèmes photovoltaïques, cette énergie répond parfaitement aux besoins des sites isolés et dont le raccordement au réseau électrique est trop onéreux.

L'énergie solaire photovoltaïque est également appelée énergie photovoltaïque. L'électricité photovoltaïque est une technologie remarquable qui transforme le rayonnement lumineux en électricité. L'effet photovoltaïque a été découvert en 1839 par le physicien français Alexandre-Edmond Becquerel. Cette filière s'est ensuite développée par la nécessité d'approvisionner en électricité solaire les satellites. [Eco info]

L'énergie solaire photovoltaïque (ou énergie photovoltaïque ou EPV) est une énergie électrique produite à partir du rayonnement solaire grâce à des panneaux ou des centrales solaires photovoltaïques. Elle est dite renouvelable, car sa source (le Soleil) est considérée comme inépuisable à l'échelle du temps humain. En fin de vie, le panneau photovoltaïque aura produit 19 à 38 fois l'énergie nécessaire à sa fabrication et à son recyclage. L'électricité, énergie non renouvelable représente un budget conséquent au sein des foyers. Pour cela, la pose de systèmes photovoltaïques peut être le remède pour faire diminuer de façon conséquente les factures d'électricité. Voici un petit condensé d'informations qu'il est nécessaire de connaître avant de se lancer dans un tel projet, sur cette base, nous parlerons de notre projet, qui est un application software pour le dimensionnement d'un système photovoltaïque [6].

1.2 Les étapes de la vie d'un projet photovoltaïque

Les étapes de la vie d'un projet photovoltaïque :

1.2.1 Préfaisabilité : La mairie a mené une réflexion afin de porter un projet solaire au sol sur le terrain le plus propice.

1.2.2 Faisabilité : Cette étape est la clé de démarrage du projet solaire. Elle consiste à mener l'ensemble des études techniques (potentiel solaire, raccordement...) et pour faire Cette Etape Il faut faire l'étude et la conception qui nécessite un logiciel spécial des systèmes photovoltaïques et que nous donne des informations utiles pour faire le projet.

1.2.3 Autorisations : La demande de permis de construire s'intègre plus globalement dans une demande dite « d'autorisation environnementale unique » qui comprend :

- un volet urbanisme (dossier permis de construire)
- une analyse du défrichement (demande de défrichement)
- une analyse des moyens de protection mis en place pour protéger la centrale vis-à-vis d'un risque de crues torrentielles (dossier loi sur l'eau)
- une analyse de l'impact de la centrale sur la faune, la flore, le paysage et son environnement de manière plus générale.

1.2.4 Financement : Suite à l'obtention des autorisations administratives et du tarif d'achat, le projet entre en financement. Il peut se faire via une levée de fonds auprès d'une banque et/ou une campagne de financement participatif.

1.2.5 Construction : Les travaux de réseau électrique sont réalisés une fois le site préparé. S'ensuit la pose des structures ainsi que des trackers solaires avant la remise en état du parc. A noter que le chantier est suivi par un écologue, expert indépendant.

1.2.6 Exploitation : L'installation sera exploitée sur une durée de 25 ans par via sa filiale spécialisée dans l'exploitation et la maintenance des parcs d'énergies renouvelables. Elle réalisera une maintenance régulière et veillera à l'entretien des clôtures, des pistes internes, et des trackers solaires. La végétation quant à elle, sera entretenue par fauche et/ou pâturage pour maintenir un couvert végétal sous les panneaux. Un entretien est également prévu à l'extérieur du périmètre clôturé de façon à ce qu'aucun arbre ne fasse de l'ombre sur les panneaux.

1.2.7 Démantèlement : A l'issue de la période d'exploitation, le site pourra être réexploité ou bien réservé à un autre usage. Dans le cas où le projet photovoltaïque n'est pas reconduit, l'ensemble de l'installation sera démantelé afin de permettre une remise en état optimale. Les panneaux photovoltaïques seront acheminés vers des centres adaptés pour recyclage, notamment via PV CYCLE, éco-organisme agréé pour la collecte et le recyclage des panneaux solaires photovoltaïques usagés. [7]

1.3 Problématique

Le dimensionnement d'un système solaire photovoltaïque a pour but de déterminer la puissance du générateur photovoltaïque et la capacité de la batterie associée à l'installation, à partir des données d'ensoleillement du site et des besoins de l'utilisateur.

Afin de faciliter le dimensionnement photovoltaïque, nous faisons appel à des logiciels associés à des calculs qui demandent suffisamment de temps d'apprentissage.

Plusieurs outils ont été conçus pour rendre le dimensionnement fiable, parmi ces outils, nous avons : les applications mobiles, l'utilisation des feuilles de calcul Excel, les applications software,... Le logiciel le plus utilisé est « PVSyst » développé initialement par un groupe de physique appliqué (GAP) à l'université de Genève.

Ces logiciels posent de sérieux problèmes à l'utilisateur, à savoir la difficulté de leur trouver des copies compatibles avec différents systèmes d'exploitation ainsi qu'une complexité en termes d'utilisation et de compréhension. Pour remédier à ces problèmes, nous avons développés une application software avec une interface moderne, dynamique et simple ; présentant des avantages de compatibilité avec tous les systèmes d'exploitation.

1.4 Étude de l'existant :

Les logiciels les plus importants utilisés pour le dimensionnement d'un système solaire photovoltaïque sont les suivants :

1.4.1 Homer PRO :

HOMER (Hybride Optimisation de Multiple Energie Ressources) est un logiciel développé pour des systèmes de production d'énergie de petites puissances. Il permet de faire des simulations de systèmes avec des énergies renouvelables et avec des énergies fossiles. Un de ses grands atouts est la possibilité de pouvoir simuler des systèmes hybrides combinant différentes sources d'énergie qu'elle soit renouvelable ou fossile.

Tout d'abord, ce logiciel permet de faire des simulations pour ensuite optimiser les systèmes et pour finalement terminer avec des analyses de sensibilité sur ces systèmes optimisés. La solution obtenue par HOMER est la solution la moins coûteuse parmi différentes combinaisons de systèmes d'énergies renouvelables, d'énergies fossiles ou de systèmes hybrides comprenant deux sources d'énergie ou plus.

Le logiciel permet de faire la simulation d'un système selon des données de gisement (solaire, éolien, diesel, etc.) en fonction d'une demande énergétique (besoins en énergie). Par la suite, il est possible d'analyser plusieurs configurations différentes pour ce même système afin d'en obtenir un système optimisé au niveau du coût. Le logiciel simule toutes les configurations demandées et donne la meilleure solution, la solution la moins chère, parmi celles-ci. Ensuite, il est finalement possible de faire des analyses de sensibilité afin de savoir si la solution trouvée reste la meilleure même s'il y a certains changements dans les différents paramètres entrés (variation du coût de la technologie, variation dans les données de gisement, etc.). Il est donc possible de faire bon nombre d'analyses avec de nombreuses configurations différentes en moins de quelques minutes de simulation. [2]

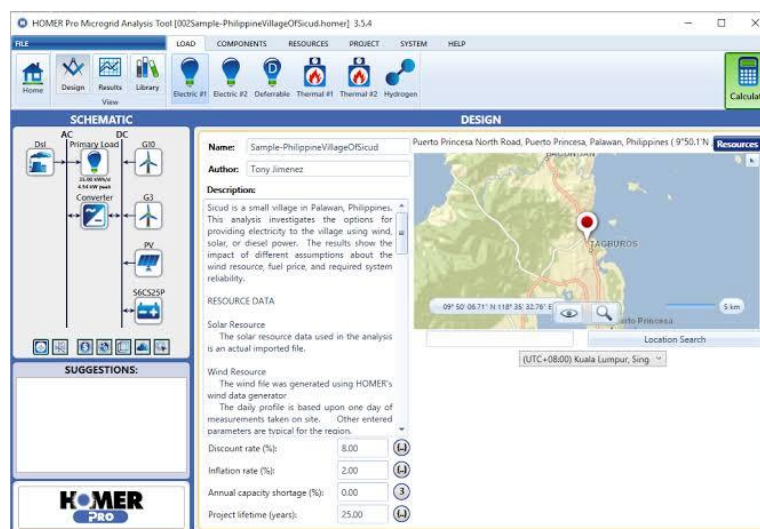


Figure 1.1: logiciel Homer pro version 3.12.

1.4.2 PV Planner :

Ce logiciel provient du fournisseur de bases de données de ressources solaires SolarGis. Il s'agit d'un logiciel basé sur le Cloud qui fonctionne sur la plate-forme SaaS (logiciel en tant que service) et affirme que leurs données sont très précises et offre une approche de validation systématique rigoureuse augmentant la fiabilité des données. Des données satellitaires précises permettent à l'utilisateur de faire une estimation du rayonnement solaire et du potentiel d'énergie photovoltaïque d'un emplacement et de générer un rapport en 14 langues. Il calcule automatiquement l'ombrage du terrain mais il n'y a pas d'options pour mettre en place des structures environnantes ou faire l'analyse de l'ombrage. Il n'est disponible qu'en version en ligne, de sorte qu'une connexion Internet est obligatoire. Permet aux entreprises d'utiliser leur API et d'accéder à Solar Gis via des applications de tiers.

Simulation basée sur la configuration facile en trois étapes. Les cartes interactives des ressources solaires haute résolution, permettent aux utilisateurs de localiser avec précision l'emplacement du site. Interface est soignée et intuitive, mais les performances de l'interface pourraient être affectées par des vitesses de connectivité Internet qui est un inconvénient des logiciels en ligne. Les données moyennes annuelles et mensuelles à long terme sont incluses dans le forfait de base. Il n'est pas possible d'importer d'autres types de données comme la NASA, Le Métronome, ou les données enregistrées par les utilisateurs. Utilise des modules génériques qui sont basés sur une moyenne de 18 modules couramment utilisés de diverses marques et les utilisateurs choisissent entre les modules, CdTe et CIS. L'onduleur est également un générique où l'utilisateur peut définir l'efficacité. Ils affirment que la variation dans la plupart des modules n'est pas supérieure à 1,22%, ce qui est inférieur à la variation du rayonnement solaire, d'où la nécessité d'avoir un module et une base de données d'onduleurs ne se pose pas [3] .



Figure 1.2 : logiciel PV planner

1.4.3 PV F-CHART :

Développé par les facultés de l'Université du Wisconsin, PV F-Chart est un programme de conception d'analyse de systèmes photovoltaïques qui utilise les données de rayonnement solaire pour calculer la production d'énergie photovoltaïque sans tenir compte des variations causées par les modules PV, les onduleurs et d'autres variables. Ce programme est destiné à des fins académiques. Programme simple qui calcule la production d'énergie PV sur la base d'un module générique et d'un onduleur. Toutes les données doivent être ajoutées manuellement et il est donc difficile de comparer rapidement les données de génération en changeant les données du module, la capacité du site ou l'emplacement. Aucune analyse d'ombrage. Ne convient pas au calcul de la puissance PV dans des conditions réelles. Un graphique simple ou des données tabulées peuvent être générés mais ne peuvent pas être exportés Très basique et pratique pour le calcul lorsque les données sont saisies manuellement pour chaque paramètre. Il s'agit d'un ensemble de feuilles de calcul prenant en charge les macros. Automatisation très limitée et aucun assistant disponible. 300 emplacements sont fournis avec le logiciel. Les données ne peuvent être ajoutées que manuellement. Aucune possibilité d'ajouter des données de module ou d'onduleur au calcul [4] .

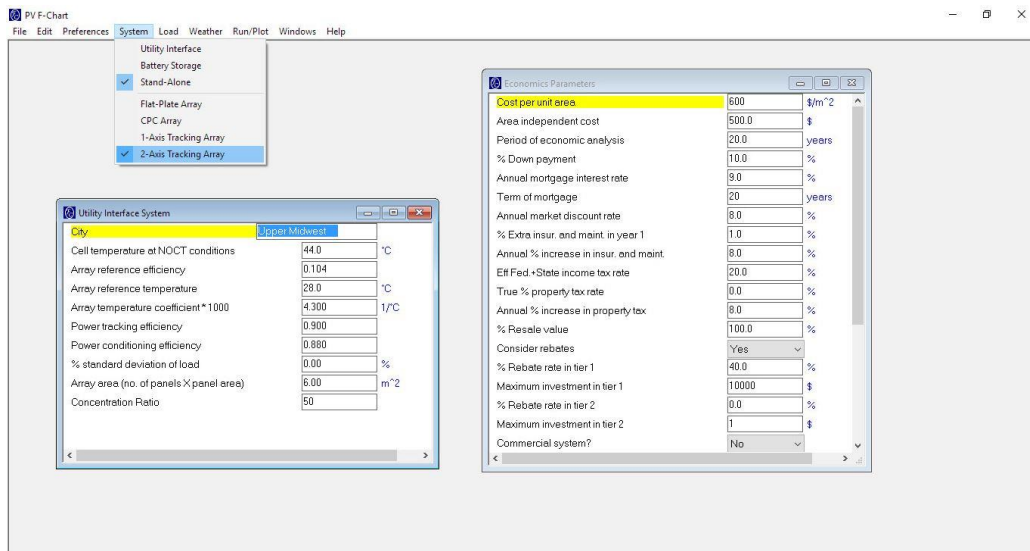


Figure 1.3 : logiciel PV F-chart

1.4.4 RET SCREEN :

RET Screen : est un logiciel de gestion d'énergies propres développé et maintenu par une Équipe principale au centre de recherche Canmet énergie situé à Varennes, avec un large Réseau d'experts de l'industrie, du gouvernement et du milieu académique procurant un support technique sur une base contractuelle ou à tâches partagées. Cette approche donne à RETScreen International un accès à un large éventail d'expertises nécessaires pour accomplir Des tâches spécialisées. L'équipe principale de RETScreen à Canmet ÉNERGIE inclut actuellement 12 personnes à temps plein, Son utilité est d'évaluer et optimiser de façon rapide

La viabilité technique et financière de projets en matière d'énergies propres et aussi mesurer et vérifier facilement la performance réelles de l'installation, ainsi que trouvé des possibilités Supplémentaires d'économie et de production d'énergies. Ce logiciel a utilisé pour la mise en

œuvre de plusieurs projets d'énergies en Canada en citant L'évaluation pluriannuelle de rendement photovoltaïque à Toronto [5].



Figure 1.4: logiciel PV RETscreen

1.4.5 PV système :

PVsyst est un logiciel PC pour l'étude, le dimensionnement et l'analyse des données des systèmes photovoltaïques complets. Il traite des systèmes PV connectés au réseau, autonome, de pompage et DC-réseau (transport public), et comprend de vastes systèmes PV et composants météo bases de données, ainsi que des outils généraux de l'énergie solaire. Ce logiciel est conçu pour les besoins des architectes, des ingénieurs, des chercheurs. Il est également très utile pour la formation pédagogique. PVsyst propose 3 niveaux d'étude du système PV, ce qui correspond à peu près aux différentes étapes du développement du projet réel :

1.4.5.1 La conception préliminaire (pré-dimensionnement) :

C'est l'étape qui préside d'un projet. Dans ce mode, les évaluations de rendement du système sont effectuées très rapidement dans les valeurs mensuelles, en utilisant seulement un très peu de caractéristiques ou de paramètres généraux du système, sans spécifier les composants du système réel. Une estimation approximative du coût du système est également disponible. Pour les systèmes connectés au réseau, et en particulier pour la construction de l'intégration, ce niveau sera l'architecte orienté, ce qui nécessite des informations sur la surface disponible, la technologie photovoltaïque (couleurs, transparence, etc.), la puissance requise ou investissement souhaité. Pour les systèmes autonomes cet outil permet à la taille de la capacité de puissance de PV et la batterie nécessaire, compte tenu du profil de charge et la probabilité que l'utilisateur ne sera pas satisfait.

Pour les systèmes, les besoins en eau donnés et une profondeur de pompage pour le pompage, et en précisant certaines options techniques générales, cet outil évalue la puissance de la pompe et PV taille du tableau nécessaire. Comme pour les systèmes

autonomes, ce dimensionnement peut être effectué selon une probabilité donnée que les besoins en eau ne sont pas respectés au cours de l'année.

1.4.5.2 Conception du projet :

Il vise à réaliser une conception approfondie du système en utilisant des simulations Horaires détaillés. Dans le cadre d'un « projet », l'utilisateur peut effectuer différents essais de simulation du système et de les comparer. Il doit définir l'orientation du plan et de choisir les composants spécifiques du système. Il est assisté dans la conception du réseau de PV (nombre de modules PV en série et parallèle), étant donné un modèle d'onduleur choisi, la batterie ou de la pompe.

Dans une deuxième étape, l'utilisateur peut spécifier des paramètres plus détaillés et analyser Les effets comme comportement thermique, le câblage, la qualité du module, inadéquation et l'angle d'incidence des pertes, l'horizon , ou ombrages partiels d'objets près de la baie, un etc... Pour les systèmes de pompage, plusieurs conceptions de système peuvent être testés et comparés les uns aux autres, avec une analyse détaillée des comportements et de l'efficacité.

Les résultats comprennent plusieurs dizaines de variables de simulation, qui peuvent être affichées dans les valeurs mensuelles, quotidiennes ou horaires, et même transférés à d'autres logiciels. La "perte Diagramme" est particulièrement utile pour identifier les faiblesses de la conception du système. Un rapport d'ingénieur peut être imprimé pour chaque exécution de la Simulation, y compris tous les paramètres utilisés pour la simulation, et les principaux résultats. Une évaluation économique détaillée peut être effectuée en utilisant les prix des composants réels, les coûts supplémentaires et les conditions d'investissement.

1.4.5.3 Bases de données :

La gestion des bases - pour données et PV composants météorologiques. Création et gestion de sites géographiques, la génération et la visualisation des données météorologiques horaires, l'importation de données météorologiques à partir de plusieurs sources prédéfinies ou à partir de fichiers ASCII personnalisés. La gestion de base de données des fabricants et des composants PV, y compris les modules photovoltaïques, onduleurs, régulateurs, générateurs, pompes, etc...

1.4.5.4 Outils :

Mesuré analyse des données : quand un système PV est en cours d'exécution et soigneusement surveillé, cette partie (situé dans les "Outils" partie) permet l'importation de données mesurées, pour afficher les tableaux et graphiques des performances réelles, et pour effectuer des comparaisons étroites avec les variables simulées. Cela donne un moyen d'analyser les paramètres réels de fonctionnement du système, et d'identifier, même très petites irrégularités. Sont inclus également quelques outils spécifiques utiles lorsqu'ils traitent avec des systèmes d'énergie solaire : tableaux et graphiques de données météo ou des paramètres de géométrie solaire, l'irradiation sous un modèle de temps clair, le comportement PV-array sous ombrages partiels ou module de décalage, des outils optimisant pour l'orientation ou de la tension, etc. La cure se définit comme l'ensemble des traitements appliqués au béton pour éviter le départ prématuré de l'eau au cours des premières heures de durcissement [8] .

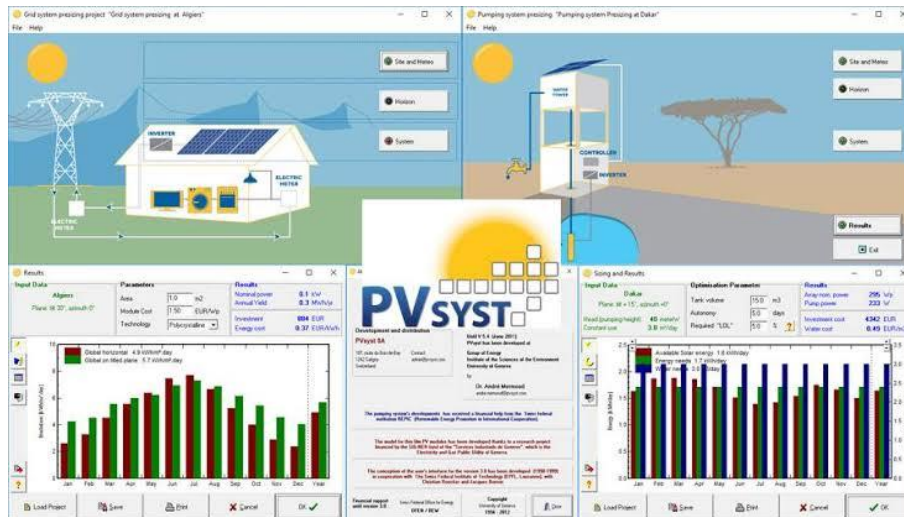


Figure 1.5 : logiciel pvsyst

1.4.6 SYSTEM ADVISOR MODEL (SAM) :

Logiciel libre développé par le US Département de l'énergie (DE) et le National Renewable Energy Laboratoire (NREL), c'est un modèle performant et financier conçu pour aider à la prise de décision (en tenant compte des données américaines et des politiques énergétiques) au stade de la planification du projet. SAM effectue des prévisions de performance et des estimations du coût de l'énergie Pour les projets d'électricité connectés au réseau en fonction des coûts d'installation et d'exploitation et des paramètres de conception du système spécifiés par l'utilisateur. Possède des préréglages pour divers types de modèles financiers utilisés aux États-Unis et l'utilisateur peut choisir l'option appropriée et fournir des entrées pour exécuter une simulation et générer des rapports sur la production, les pertes et les données financières. N'offre pas d'analyse d'ombrage mais peut importer ces données depuis PVsyst.

Nécessite beaucoup de saisie manuelle de données et si vous êtes une personne qui n'a pas d'expérience en conception de systèmes photovoltaïques, vous pourriez trouver cela un peu écrasant. Télécharge les données de la base de données nationale sur le rayonnement solaire Des États-Unis qui couvre de nombreuses villes du monde entier. Permet l'importation de données TMY2, TMY3 et EPW. Une grande base de données de modules et d'onduleurs répertoriés par CEC (Californie Electricité Commission).

1.4.7 Solar pro :

C'est le seul logiciel de la liste MNRE qui offre un calcul minute par minute, ce qui en fait l'un des logiciels les plus précis et également le seul à offrir une interface utilisateur 3D interactive qui permet de visualiser l'installation du système PV et de voir en temps réel modifications de l'ombrage et de la production d'énergie en déplaçant des objets dans l'espace 3D. Offrant des niveaux élevés de précision de la production d'énergie PV grâce à sa formule de calcul minute par minute unique qui prend une lecture cumulée de kilowatts au lieu de la lecture moyenne habituelle du kilowattheure prise par la plupart des autres

logiciels. En outre, son analyse interactive détaillée de l'ombrage proche permet à un utilisateur d'identifier les zones spécifiques qui sont affectées et d'apporter les modifications nécessaires à la conception pour améliorer la génération. Il convient également de mentionner la fonction Carte qui aide à dessiner une mise en page de site à l'aide d'images satellites au sol. L'une des interfaces utilisateur les plus conviviales.

Un assistant de simulation guide l'utilisateur pour configurer rapidement une installation, qu'il s'agisse d'un projet de niveau toit ou mégawatt. L'environnement CAO 3D permet à un utilisateur de créer de manière interactive l'environnement autour de lui en faisant glisser et déposer des objets 3D. Des animations précises affichent l'effet de l'ombrage sur les modules et les utilisateurs peuvent ajuster la position des objets venant dans le chemin du Soleil et voir immédiatement le résultat sur les modules affectés. La fonction de carte intégrée permet au concepteur de gagner du temps pour effectuer une Étude visuelle du site et un calcul de surface [9].

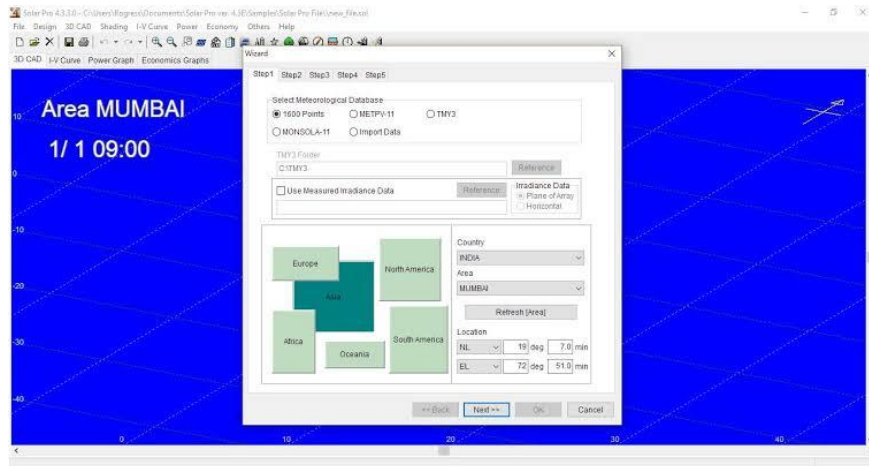


Figure 1.6 : logiciel Solar pro



Figure 1.7: système advisor

1.5 Comparaison entre pvsyst et Homer pro :

Le tableaux suivant représente la comparaison entre logiciel pvsyst et Homer pro :

Tableau 1.1 : Comparaison entre PVsyst et Homer pro.

Logiciel de dimensionnement	Pvsyst	Homer pro
Les données géographiques (choix de site)	1-base de données intégrer (200site) 2-on peut définir manuellement avec la base de données de NASA	1-trouve directement à partir d'une carte géographique intégrée à celui-ci, à partir (google maps)
Les besoins	Dépend du matériel du client	Dépend de la quantité d'énergie de charge électrique
L'étude	1-Etudie Seulement le système photovoltaïque 2- Étudier seulement les changements d'irradiation 3-Chaque fois qu'il nous donne des alertes d'erreur. 4- étudier les pertes	1-Les Système des énergies renouvelables (éolien ,thermique,photovoltaïque). 2-Les énergies fossiles . 3- Il étudie les changements dans plus d'un facteur exemple (batterie)
résultats(donne les rapports vous trouverez):	1-il donne une seule résultat	1-il donne beaucoup des résultats iu des suggestions 2-Il concentre ses études sur l'aspect économique 3-il donne la solution de bon prix (moins chère)
Autre	Il a une orientation des panneaux photovoltaïques.	1-Il ne fonctionne que sur de petits systèmes de moins de 1Mw 2-Ne ne donne pas de résultats précis pour l'énergie éolienne. 3-Déterminer la durée de vie du projet

1.6 Cahier de charge

1.6.1 Le projet : Dimpower est une application software de dimensionnement d'un système photovoltaïque sur tous les systèmes exploitation (mac, Windows, linux) son but est de faire un dimensionnement selon une manière simple et rapide.

1.6.2 Les objectif :

- Connaître les données de l'irradiation
- Connaître le numéro des équipements solaire nécessaire

- connaitre la caractéristique géographique du site
- connaissance des valeurs : énergie journalier et l'énergie produit de système
- connaissance des valeurs : l'énergie consommée et puissance utile
- le résultat peut être téléchargé

1.6.3 La cible : Les étudiants, les entreprises, les chercheurs

1.6.4 Les contraintes générales :

- **temps de développement :** Le temps alloué pour réaliser l'application est de 3 mois.

1.6.5 Les livrables attendus : Fichier PDF du résultat

1.7 Conclusion :

Ce premier chapitre définit tout d'abord le système photovoltaïque et leur utilité dans nos vies quotidiennes et la nécessité de faire la conception et des études pour faire développement de projet photovoltaïque avec des logiciel spéciales pour faciliter notre travaille. Ensuite nous avons parlé du problématique qui nous a poussé à concevoir une application software multiplateforme pour le dimensionnement d'un système photovoltaïque et nous présentons a sa cahier de charge, basé sur notre étude de plusieurs applications utilisées. Dans le prochain chapitre, nous allons passer à la conception de notre application software.

Chapitre 2 :

Conception de l'application software

2.1 Introduction :

De nos jours, chaque entreprise d'énergie solaire recherche des méthodes plus adaptées et plus efficaces pour réaliser leur dimensionnement de ses projets d'installation photovoltaïque en moins de temps possible. Ainsi, dans une entreprise il est nécessaire de passer par un bon logiciel informatique, il en existe beaucoup qui sont adaptés au différent utilisateur et aux d'entreprise.

Dans ce chapitre, nous allons présenter l'application existante qui est une application Software multiplateforme pour le dimensionnement d'une installation photovoltaïque, tout en détaillant les fonctionnalités de notre application, ou nous verrons la Description de l'organisation des différentes interfaces ; dans lequel nous avons utilisé la méthode de « UML » qui permettra de bien représenter les aspects fonctionnels statique et dynamiques de notre projet par la série des trois diagrammes et le diagramme de communication qu'il offre, suivi d'une phase « conception de l'interface » s'appuyant sur la partie précédent de « UML », pour faire ce conception passée par deux étapes : Maquettage avec adobe Illustrator et Réalisation de programme selon la méthode SPA(single page application).

2.2 Application web monopage :

Une application web monopage (single page application SPA en anglais) est une application web dont l'ensemble des éléments nécessaires (contenu, présentation : HTML, image, css, et partie applicative :JavaScript) se trouve dans un unique fichier HTML, le terme a été introduit par Steve yen en 2005.

L'enregistrement en local de page définissant une application web monopage et la possibilité de continuer à l'exécuter en local est l'une des propriétés importantes des applications web monopage qui les distingue des applications web standards qui reposent sur l'existence d'un serveur http avec lequel elles échangent données, continuations applicatives et interfaces. Quand les application web monopage gèrent des données et permettent de les modifier, pour conserver ces données modifiées, il est nécessaire que ces application modifient leur code, c'est-à-dire : elles doivent être capable de se modifier pour que la sauvegarde locale de l'état modifié de l'application (dont les données) soit persistant [10].

2.3 Présentation de l'application

2.3.1 Description des fonctionnalités de l'application :

Notre application software présenté (5) grandes Etapes :

- **Etape 1 :** le choix de type de l'installation il voudrait réaliser ensuite le système offre plusieurs possibilités installations avec stockage ou sans stockage ou bien système connecté aux réseaux au système hybride, un fois l'utilisateur choisit le système autonome logiciel passer à l'étape 2 mais pour les autres choix logiciel donne chaque fois un message d'erreur, l'installation avec stockage implique le dimensionnement des batteries, des onduleurs des panneaux au des régulateur,
- **Etape 2 :** le choix de site où l'installation sera fait, cette étape permet à l'utilisateur de définir le site géographique de l'installation ce choix permet à notre application de pouvoir la façon automatique définir l'irradiation annuelle de site géographique,
- **Etape 3 :** le choix de composant dans cette étape l'utilisateur pourra définir clairement tout ce qu'il a besoin d'alimenter en électricité à cet effet un tableau lui sera proposée et il pourra choisir tous les différents composants (lampes , machine à laver , réfrigérateur , télévision ..) il pourra ajouter autre équipement qui n'avaient pas prédéfini , l'utilisateur doit renseigner également la puissance unitaire de chaque composant et leur temps d'utilisation , ce qui va permettre au logiciel d'établir l'énergie consommée totale et la puissance utile ce qui lui permet ensuite de calculer le nombre de panneaux et le nombre de batterie .
- **Etape 4 :** dans cette étape l'utilisateur permet choisir les types des panneaux et des batteries à partir d'une liste intégrée.
- **Etape 5 :** cette étape donne l'affichage de résultat (le nombre de panneaux et l'utilisateur de télécharger son résultat

2.3.2 Diagramme de communication entre différentes interfaces :

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet (voir l'historique d'UML sur la figure 2-1). UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression de besoin jusqu'au codage. Dans ce cadre, un concept appartenant aux exigences des utilisateurs projette sa réalité dans le modèle de conception et dans le codage. Le fil tendu entre les différentes étapes de construction permet alors de remonter du code aux besoins et d'en comprendre les tenants et les aboutissants [11].

Historique d'UML

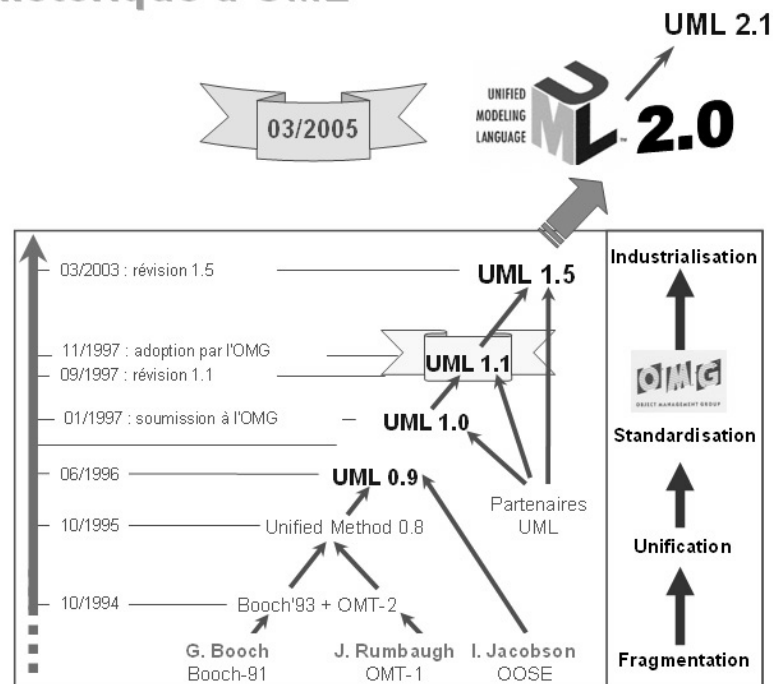


Figure 2.1 : Historique d'UML

UML compose des treize types de diagrammes, mais nous utilisons seulement trois, en insistant particulièrement sur les diagrammes de classes et les diagrammes de séquence et diagramme de cas d'utilisateur (figure 2.2). Cette limitation volontaire permet une réduction significative du temps d'apprentissage de la modélisation avec UML, tout en restant largement suffisante pour la plupart des projets.

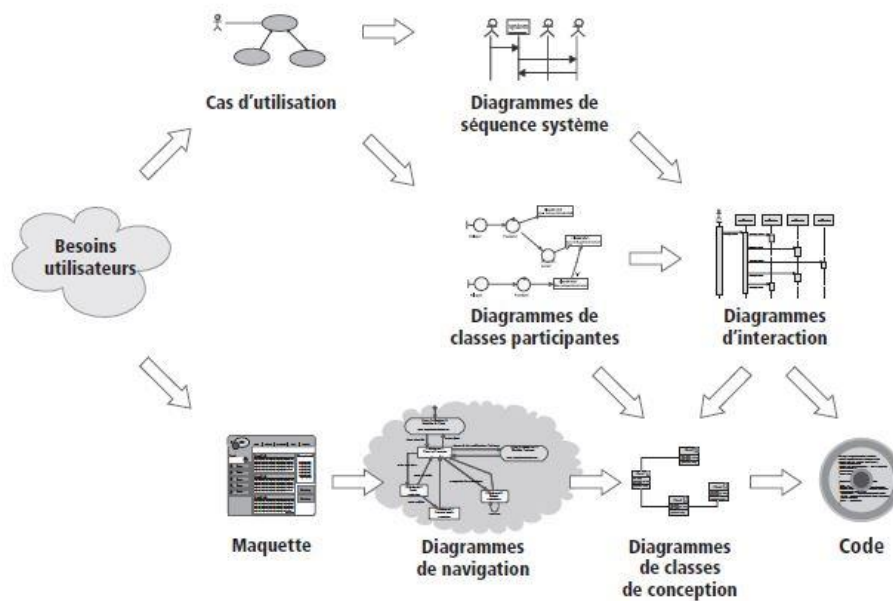


Figure 2.2 : Schéma complet du processus de modélisation d'une application web.

2.3.2.1 Diagramme dynamique :

2.3.2.1.1 Diagramme de cas d'utilisateur :

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel.

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (Humane ou Machine) et un system. Il est une entité significative de travail Dans un diagramme de cas d'utilisation il existe des acteurs (acteurs) qui interagissent avec des cas d'utilisation (use case) UC. Les use case permettent de structurer les besoins des utilisateurs et les objectifs du système [12] . Comme l'illustré la (figure 2.3) il y a deux acteurs dans notre application software qui donner et gérée les l'information de système (l'utilisateur et google maps).

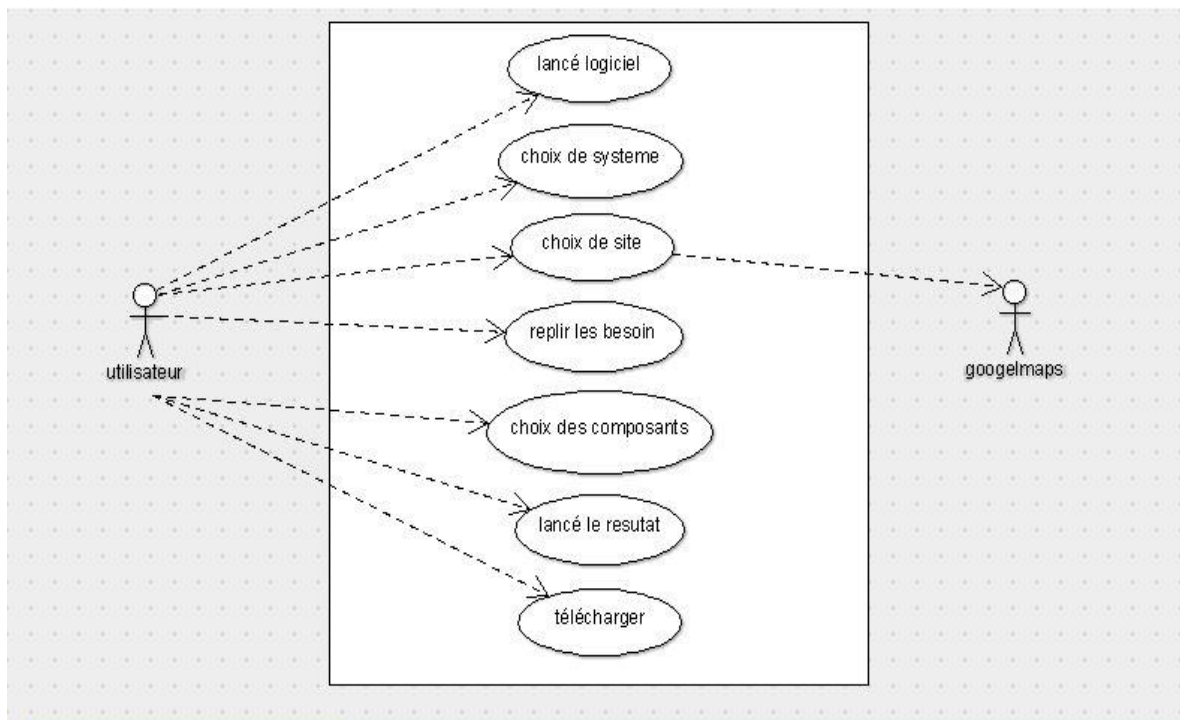


Figure 2.3 : schéma de diagramme de cas d'utilisateur par logiciel ArgoUML (réalisation personnelle)

L'utilisateur installe et entre l'information de l'installation photovoltaïque (choix de système, choix de site, les besoins, choix de composant), google maps donne les caractéristiques géographiques de leur site choisi. L'utilisateur peut télécharger le résultat pour contacter un fournisseur d'équipement solaire photovoltaïque.

2.3.2.1.2 Diagramme de séquences :

Le diagramme de séquence est une variante du diagramme de collaboration. Par opposition aux diagrammes de collaboration, les diagrammes de séquence possèdent intrinsèquement une dimension temporelle mais ne représentent pas explicitement les liens entre les objets. Ils privilégient ainsi la représentation temporelle à la représentation spatiale et sont plus aptes à modéliser les aspects dynamiques du système. En revanche, ils ne tiennent pas compte du contexte des objets de manière explicite, comme les diagrammes de collaboration.

Le diagramme de séquence permet de visualiser les messages par une lecture de haut en bas. L'axe vertical représente le temps, l'axe horizontal les objets qui collaborent. Une ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie. [13] Selon le (Figure 2.4) le diagramme de séquence de notre application contient les objets suivant (l'utilisateur, système autonome, autre système, google maps, besoin, panneaux, batterie, résultat). Si l'utilisateur choisir le système autonome le system passé a les autres étapes de dimensionnement sinon (le cas d'un autre système) logiciel affiché un message d'erreur et bloque les autre étapes suivant.

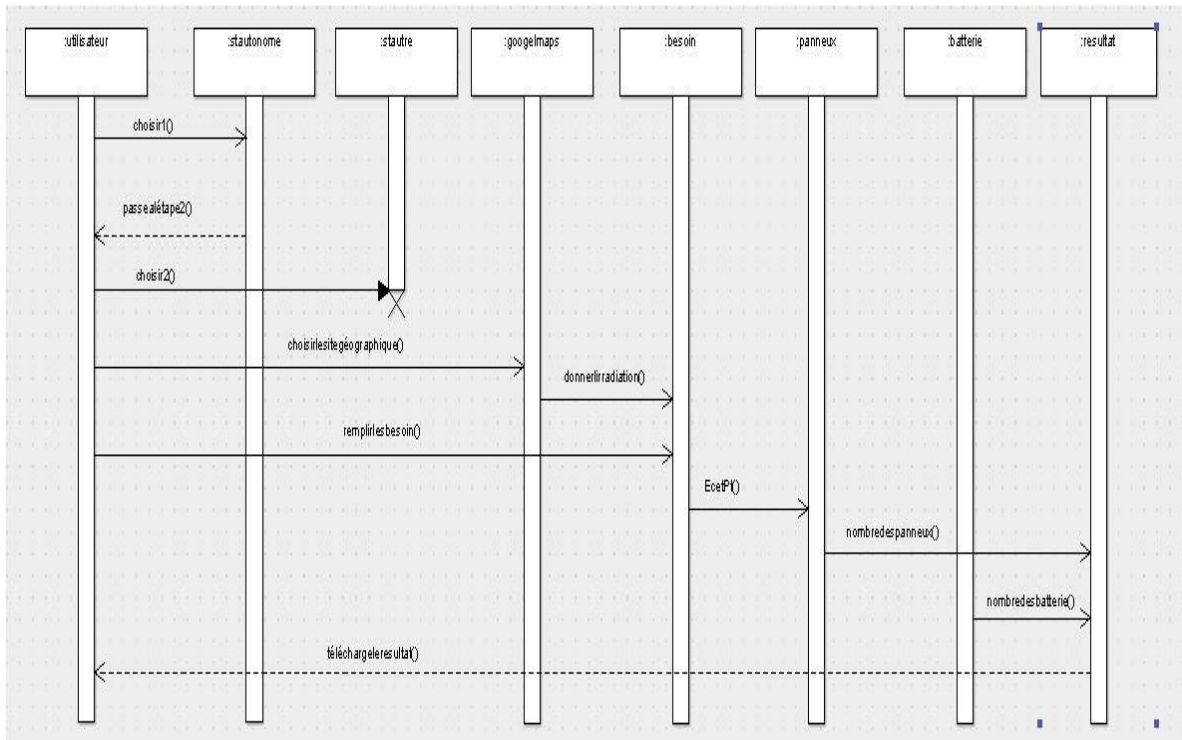


Figure 2.4 : schéma de diagramme de séquence par logiciel ArgoUML (réalisation personnelle)

2.3.2.2. Diagramme structure (statistique) :

2.3.2.2.1 Diagramme de classe :

Le diagramme de classe est basé sur les notions fondamentales de classe d'objet et d'associations[14], en effet, ce diagramme (figure 2.5) permet de donner la représentation statistique de l'application développée, cette représentation est centrée sur les concepts de classe et d'association, dans notre application notre diagramme des classes contient les classes : système d'installation, site géographique, panneau, batterie, besoin.

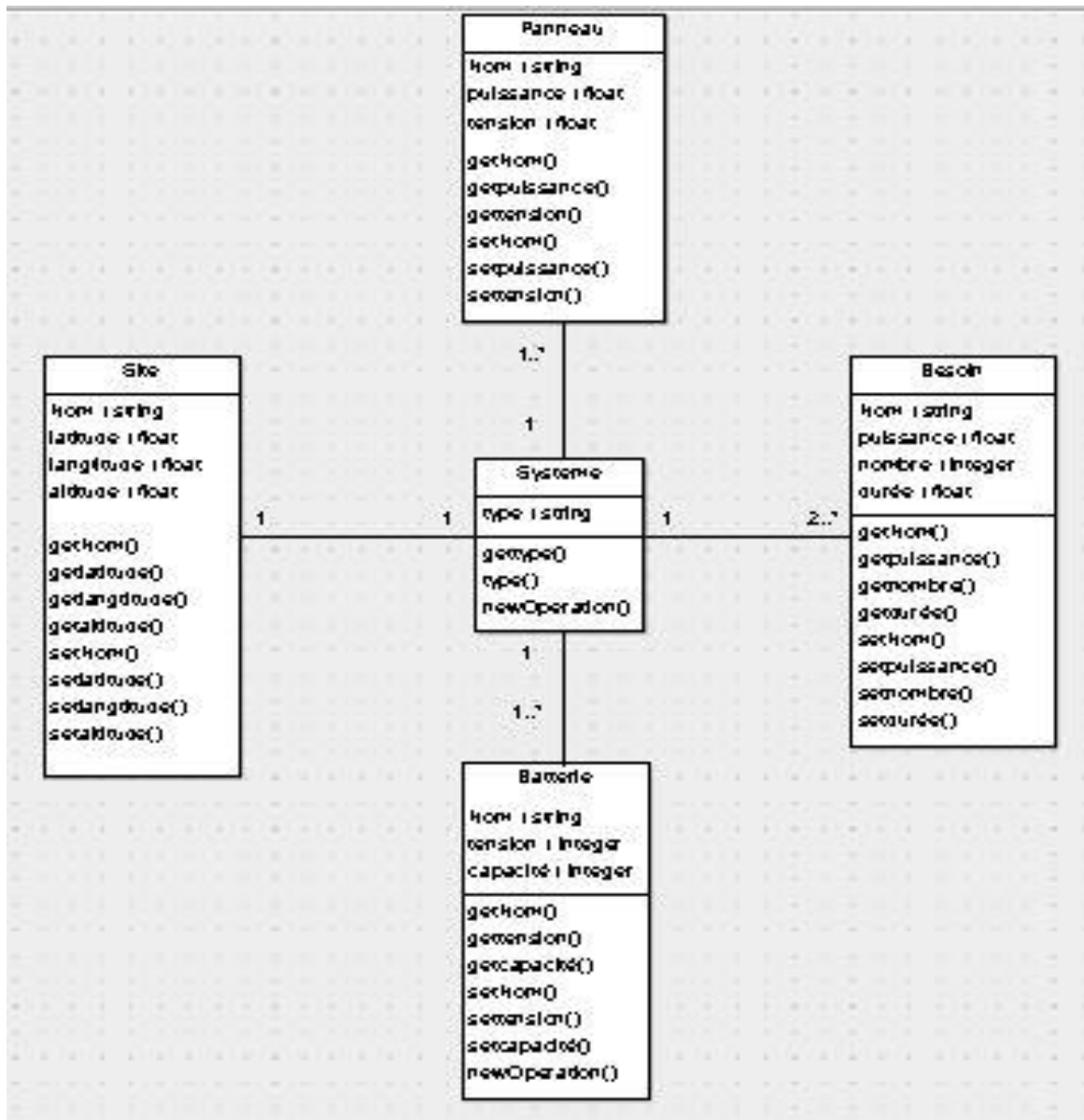


Figure 2.5 : schéma de diagramme de classe par ArgoUML (réalisation personnelle)

2.3.2.3. Diagramme d'activité :

Le figure 2.6 représenter Diagramme de communication entre diffèrent interface, lorsque nous ouvrons l'application, cela nous amène a une « page man », nous trouvons deux possibilité (choix d'un système autonome ou bien autre système), le choix système autonome nous amène a d'autres page de l'application jusqu'à ce que nous atteignons la page de résultat .

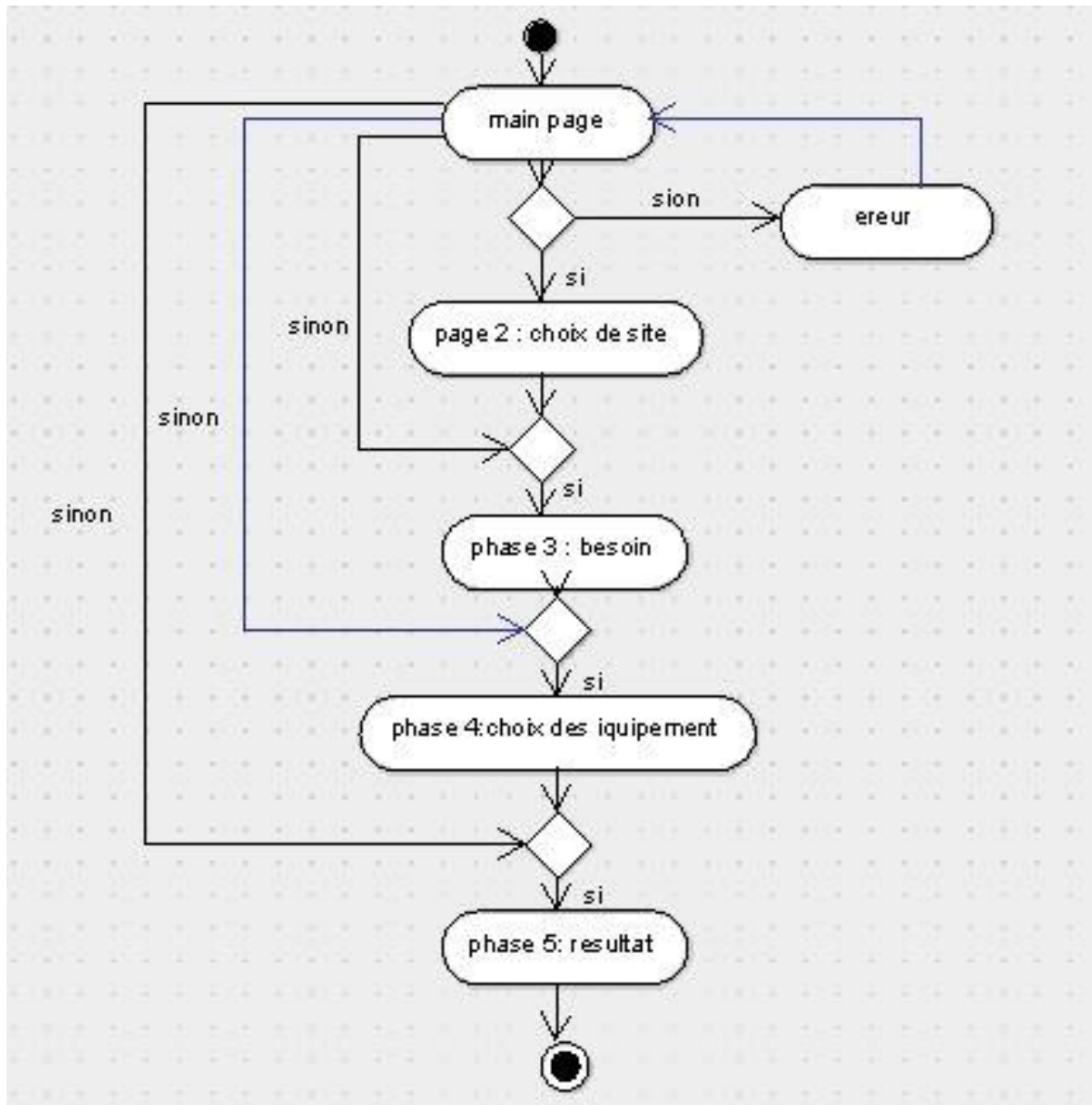


Figure 2.6: schéma de diagramme d'activité par ArgoUML (réalisation personnelle)

2.3.3. Les données nécessaires de l'application :

2.3.3.1 Les entrés :

Utilisation d'Input: navigation avec la carte géographique, modification des données, y compris le remplissage des formulaires, l'affichage des données existantes et le téléchargement des résultats

- 1-les besoin
- 2-choix d'un système
- 3-choix d'un site géographique

2.3.3.2 Les sorties

1-les valeurs :

Énergie consommée : c'est la quantité utilisée par nos panneaux solaires

Nombre de panneaux en parallèle/en série

Nombre batterie.

2-les graphes :

Énergie produit de système leur caractéristique, le nombre de batterie, l'onduleur et le régulateur avec l'irradiation annuelle et l'énergie produit et leur graphes)

2.4. Structure de l'interface d'utilisateur (UI) :

Dans cette partie, nous allons présenter les différentes phases de conception de notre application en mentionnant des imprime écrans de chaque phase .

2.4.1 La charte graphique de l'application :

L'idée de la charte graphique de notre application est un globe terrestre composé de cellules solaires divisées en deux parties qui expriment l'union de la terre avec le soleil, une ligne courbe entre les deux parités pour donner du mouvement au logo. Pour réaliser cette idée nous avons tracé une cercle et nous l'avons dupliqué(Alt) plusieurs fois , puis nous avons appliqué chaque fois l'option (masque) , ensuit avons fait les arondes pour tous les formes qui exprime les cellules solaire , enfin nous avons coloré le résultat selon les trois couleur choisi et écrit le nom de notre application au-dessous de logo (figure 2.7 et figure 2.8).

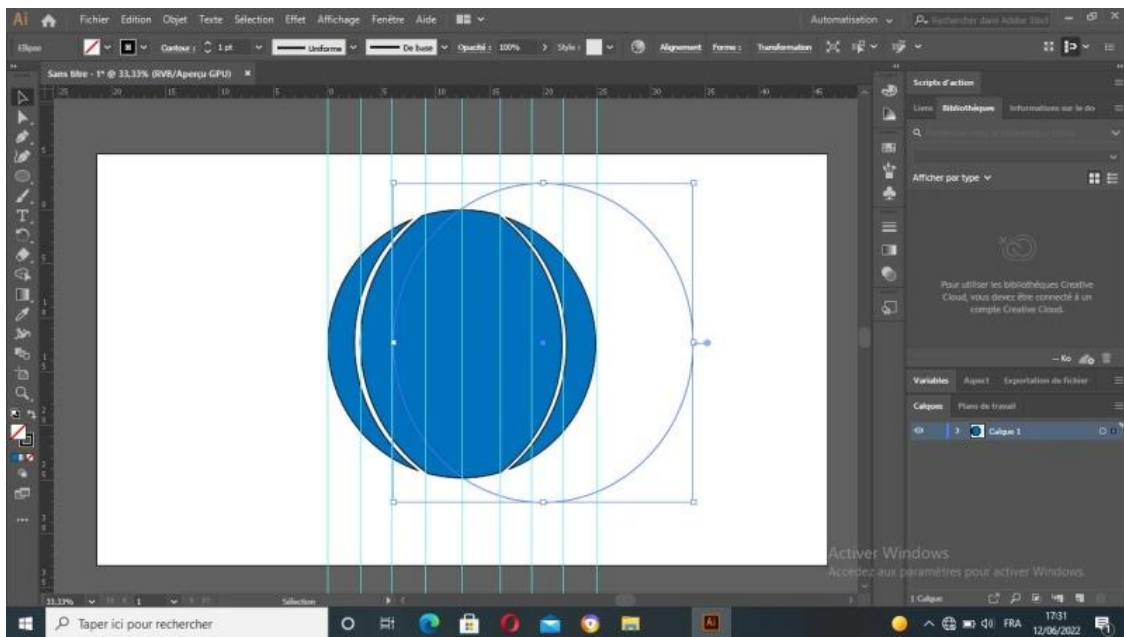


Figure 2.7 : l'idée générale de la réalisation de logo



Figure 2.8 : logo de l'application software

2.4.2 Maquettage avec adobe Illustrator :

On a ouvert un nouveau document dans logiciel « Adobe Illustrator cc 2019 » selon les caractéristique suivant (figure 2.9) :

Taille : largeur = 1366 pixel, hauteur = 768 pixel.

Mode colorimétrique : couleur RVB.

Résolution : 72 ppp.

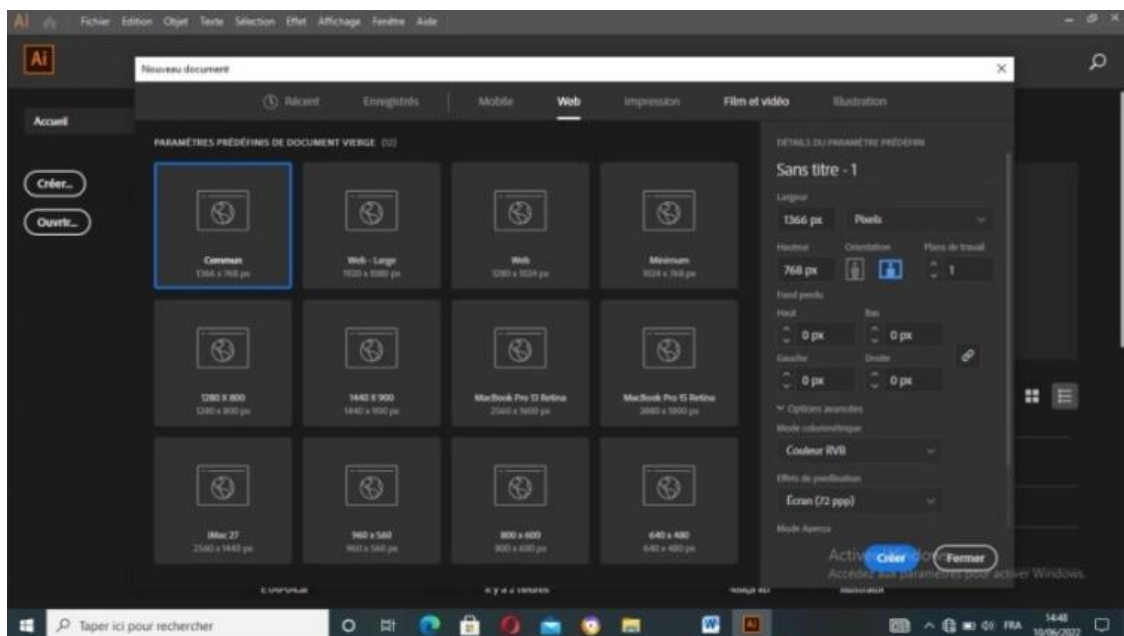


Figure 2.9 : capture de nouveau document dans Adobe Illustrator

Ensuite on a ajouté un nuancier de couleur suivant (figure 2.10) :

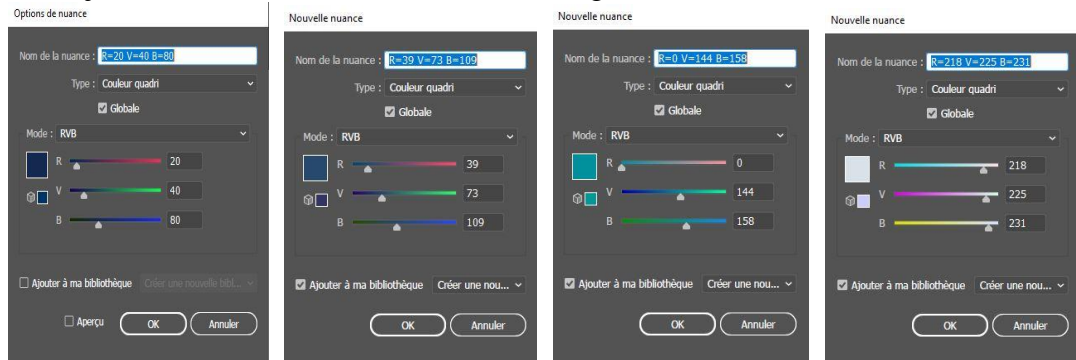


Figure 2.10 : capture de choix de couleur dans Adobe Illustrator

Puis on a fait l'illustration de toutes les phases de l'application comme la suit :

- **Phase 1 :** avec l'outil rectangle, nous avons tracé une forme et nous l'avons colorée avec l'outil pipette on bleu, ensuite nous nous avons ajouté une rectangle et dupliquer quatre fois avec la touche (Alt) pour la présentation des différents systèmes photovoltaïque puis on a ajouté deux rectangles (horizontal et vertical) pour les barres d'option et les icônes en blanche qui nous les avons pré-dessinés (Figure 2.11).

Pour dessiner la carte cotée (Figure 2.12) nous l'avons ajoutée un petit rectangle puis ajoutée l'image de notre logo et la croix au-dessus.

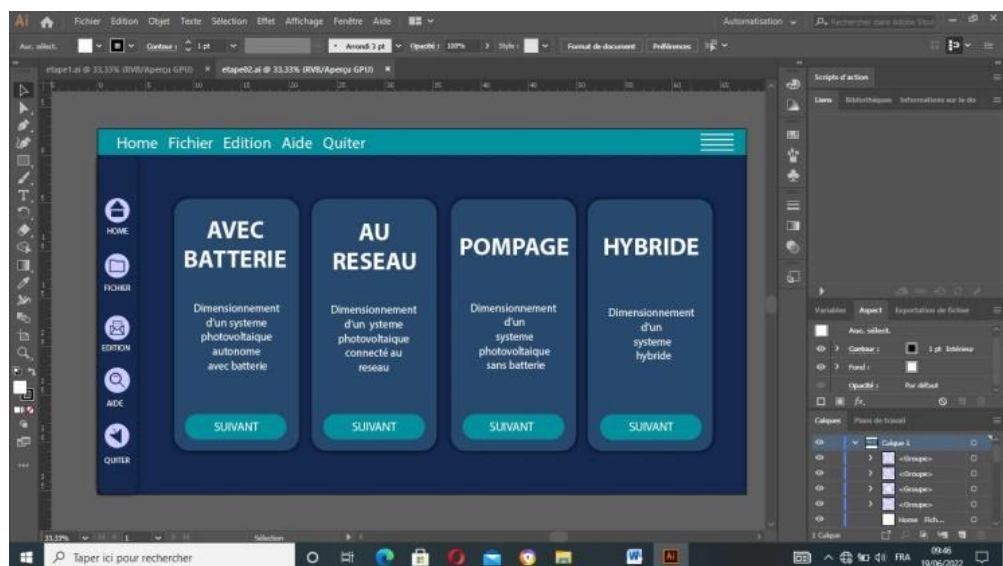


Figure 2.11 : capture de 1^{ème} phase de l'application par Illustrator

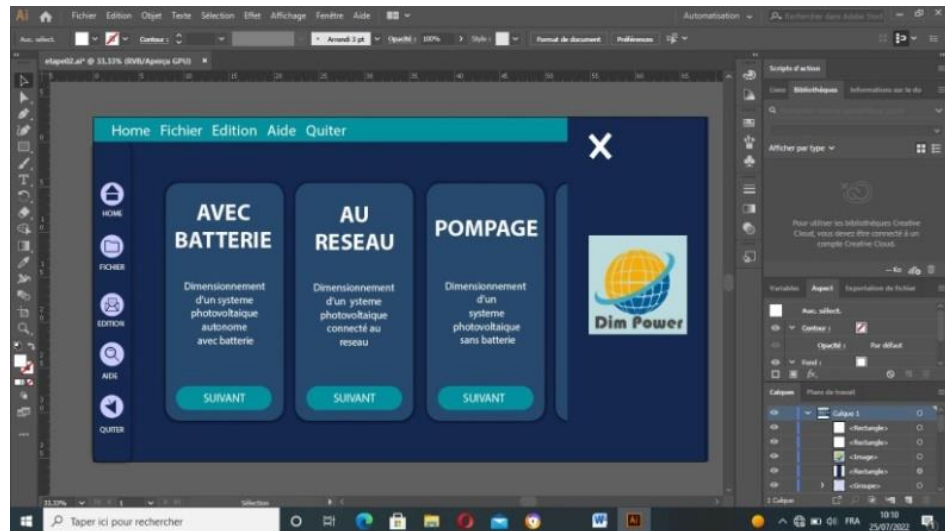


Figure 2.12 : capture de menu latéral

- **Phase 2** : pour la conception de carte géographique (Figure 2.13), nous avons ajouté une image d'une carte qui est téléchargé à l'avance et on a ajouté au dessus un rectangle blanche puis nous avons sélectionné les deux et créé un masque pour le fusionnement .

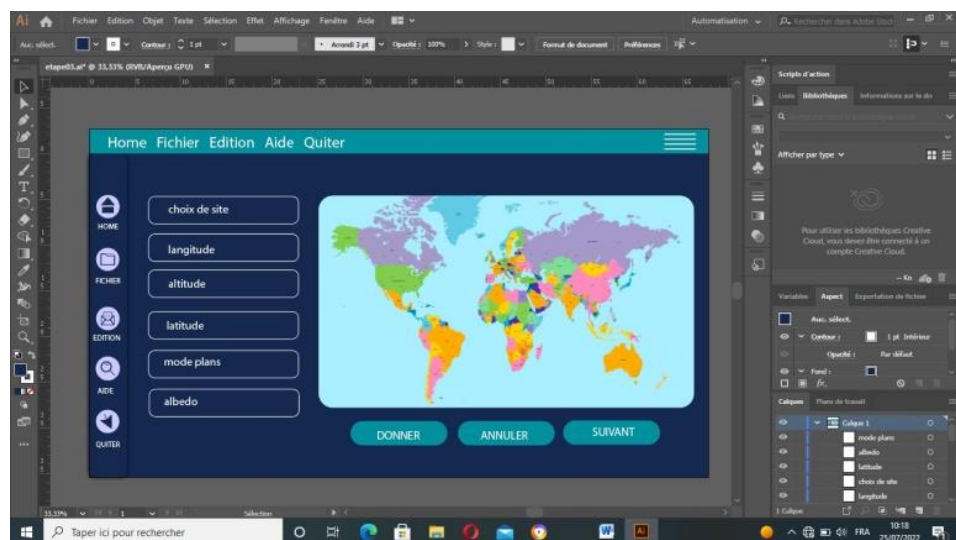


Figure 2.13 : capture de 2^{ème} phase de l'application par Illustrator

- **Phase 3** : la figure 2.14 représente le schéma de phase 4 , nous avons tracé un rectangle bleu et déminé l'opacité ensuite , en a ajouté des lignes avec l'utile traces pour séparer les colonnes de table de besoin puis nous avons ajouté les textes et les nombres .

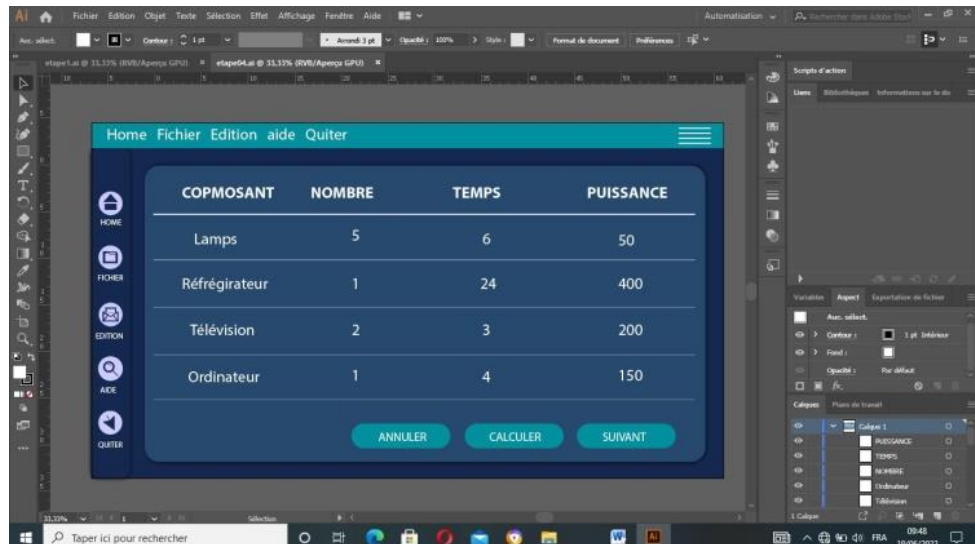


Figure 2.14 : capture de 3^{ème} phase de l'application par Illustrator

- **Phase 4 :** dans la figure 2.15 il y a deux rectangles pour les choix des types de panneaux et batteries, pour leur conception, nous avons utilisé l'outil rectangle pour tracer deux formes et nous avons ajouté les arondes de chaque rectangle avec l'utile sélectionné et la touche (Ctrl).

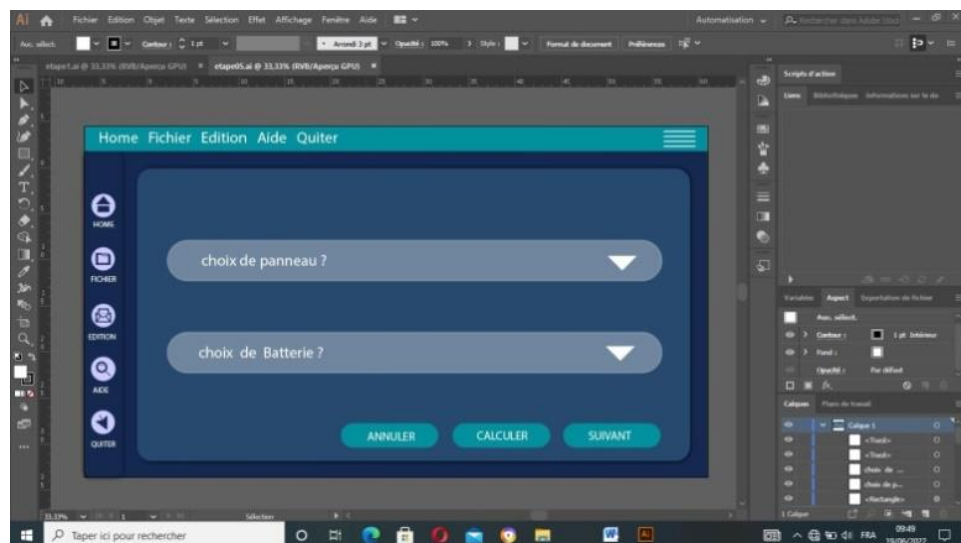


Figure 2.15 : capture de 4^{ème} phase de l'application par Illustrator

- **Phase 5 :** la figure 2.16 présente un exemple de résultat finale d'un projet de dimensionnement que nous l'avons concept par logiciel Illustrator, les deux rectangles dans la figure 2.16 donne (table de besoin, texte, et deux graphe qui représente l'irradiation annuelle et énergie produit).

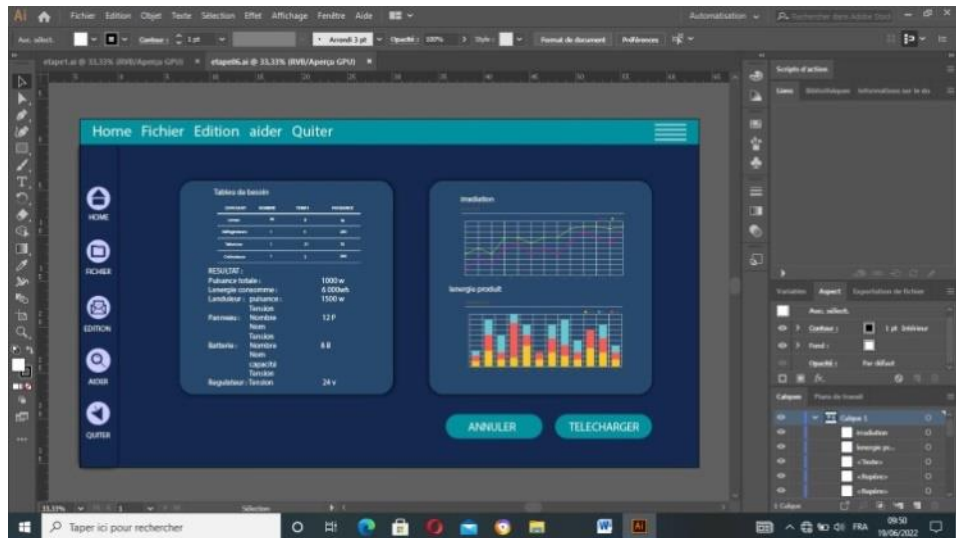


Figure 2.16 : capteur de 5^{ème} phase de l'application par Illustrator

2.5.3 programme d'arrière-plan de l'application :

Après l'analyse des besoins et la définition de la méthodologie de conception nous programmons la plupart de nos conceptions dans un environnement approprié, en utilisant une méthode qui permet à l'application de fonctionner sur différents systèmes d'exploitation, à partir d'un langage de développement web, La conception de notre application web fait selon la méthode SPA (single page application) comme illustré dans les étapes suivant :

- **Bootstrap** : Ce document contient des fichiers de bootstrap (JavaScript, css, jquery) qui permet l'utilisateur d'utiliser l'application en hors connexion.

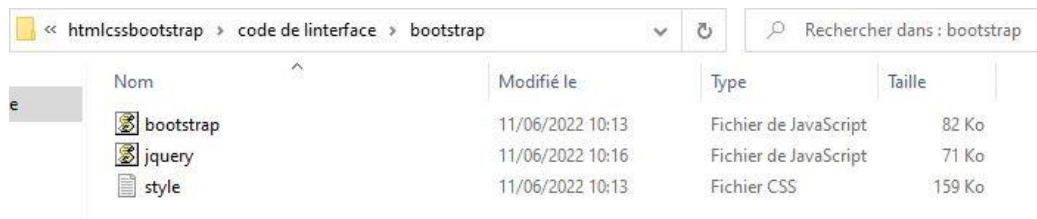


Figure 2.17: capture de fichier bootstrap

- **Style.css** : ce fichier montre les codes css (couleur, position, nom) utilisé pour les classes de HTML.

```
C:\Users> j\ > Desktop > zakia > MAMEMOIRE20202021 > App-win32-x64 > resources > app > # style.css
1
2   color: □#ffffff;
3   font-size:20px;
4   text-decoration:none;
5 }
6 .card-deck{
7   height:90%;
8   width:98%;
9 }
10 }
11
12 #c3{
13   margin-left:13px;
14   margin-top:40px;
15 }
16 .card-body{
17   text-align:center;
18   margin-top:60px;
19 }
20 }
21 #d1{
22   color: ■#000000;
23   text-align:left;
24 }
25 #zzz{
26   color: □#ffffff;
27 }
28 }
29
30 #B{
31   color: □#ffffff;
```

Figure 2.18 : capture 1 de code css

```
<> index.html # style.css
C: > Users > ji > Desktop > zakia > MAMEMOIRE20202021 > App-win32-x64 > resources > app >
30 #B{
31     color: #ffffff;
32     text-align: center;
33 }
34 }
35 #L{
36     color: #ffffff;
37     text-align: center;
38 }
39 }
40 #pp{
41     color: #ffffff;
42     text-align: center;
43 }
44 }
45 #ee{
46     color: #ffffff;
47     text-align: center;
48 }
49 }
50 #ii{
51     color: #ffffff;
52     text-align: center;
53 }
54 }
55 #tt{
56     color: #ffffff;
57     text-align: center;
58 }
59 #H{
60     color: #ffffff;
```

Figure 2.19 : capture 2 de code css

```
<> index.html # style.css
C: > Users > ji > Desktop > zakia > MAMEMOIRE20202021 > App-win32-x64 > resources > app > # st
59 #H{
60     color: #ffffff;
61     text-align: center;
62 }
63 }
64 }
65 #z{
66     color: #ffffff;
67     border-style:none;
68     text-align:center;
69 }
70 .section5{
71     margin-left:200px;
72 }
73 #klm{
74     margin-top:13px;
75     margin-left:190px;
76     height:92%;
77     width:95%;
78 }
79 #x{
80 }
81     margin-left:90px;
82     margin-top:13px;
83     height:90%;
84     width:100%;
85     text-align:center;
86     border-radius:20px;
87 }
88 p{
89     color:#ffffff;
```

Figure 2.20 : capture 3 de code css

```
index.html # style.css
C:\Users\ji\Desktop> zakia > MAMEMOIRE20202021 > App-win32-x64 > resources > app >
88 p{
89   color: #ffffff;
90   text-decoration:none;
91 }
92 #M{
93
94   margin-left:400px;
95   margin-top:40px;
96
97
98 }
99 #R{
100   border-radius:20px;
101
102   width:100%;
103   color: #ffffff;
104 }
105 #h{
106   border-radius:20px;
107
108   width:100%;
109   color: #ffffff;
110 }
111 .sidenav{
112   height: 90%;
113   width: 0;
114   position: fixed;
115   z-index: 1;
116   top: 0;
117   right:0;
118   background-color: #142850;
```

Figure 2.21: capture 4 de code css

```
index.html # style.css
C:\Users\ji\Desktop> zakia > MAMEMOIRE20202021 > App-win32-x64 > resources > app > #
109   color: #ffffff;
110 }
111 .sidenav{
112   height: 90%;
113   width: 0;
114   position: fixed;
115   z-index: 1;
116   top: 0;
117   right:0;
118   background-color: #142850;
119   overflow-y:hidden;
120   transition: 0.5s;
121   padding-top:50%;
122 }
123
124 .sidenav a{
125   padding-top: 8px 8px 8px 32px;
126   text-decoration: none;
127   font-size:80px;
128   color:#ffffff;
129
130   transition: 0.3s;
131 }
132 h1{
133   color: black;
134 }
135
136 .sidenav .closebtn{
137   position: absolute;
138   top:0;
139   right:25px;
```

Figure 2.22 : capture 5 de code css


```

index.html # style.css
C:\Users> ji > Desktop > zakia > MAMEMOIRE20202021 > App-win32-x64 > resources > app > # style.cs
125 padding-top: 8px 8px 8px 32px;
126 text-decoration: none;
127 font-size: 80px;
128 color: #ffffff;
129
130 transition: 0.3s;
131 }
132 h1{
133 color: black;
134 }
135
136 .sidenav .closebtn{
137 position: absolute;
138 top: 0;
139 right: 25px;
140
141 margin: right;
142 }
143 }
144 #main{
145 font-size: 30px;
146 cursor: pointer;
147 margin-left: 1270px;
148 margin-top: 3px;
149 position: fixed;
150 color: #ffffff;
151 }
152 }

```

Figure 2.23 : capture 6 de code css

Par la suite nous allons présenter les codes « HTML, JavaScript et Vues.js » utilisé pour réaliser la conception de l’interface de chaque phase de notre application software.

- **Bar d’option :**

Nous l’avons programmé le code de les deux bars d’option au dessus et à gauche un seul Foix.

```

index.html # style.css
C:\Users> ji > Desktop > zakia > MAMEMOIRE20202021 > App-win32-x64 > resources > app > index.html > html
1 <html>
2 <head>
3 <meta charset="UTF-8">
4 <!-- https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP -->
5
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7 <title> Application software </title>
8 <!-- Bootstrap -->
9 <link rel="stylesheet" href="./bootstrap/style.css" >
10 <!-- css -->
11
12 <link href="./style.css" rel="stylesheet">
13 <script src="./vue@2.7.4.js"></script>
14 </head>
15 <body>
16 <!-- container -->
17 <div style="height: 100vh; display: flex; flex-direction: column;">
18 <!-- menue lateral -->
19 <div id="mysidenav" class="sidenav">
20 <a href="#" class="closebtn" onclick="closeNav()">
21 &times;
22 <br>
23 <br>
24 
25 </br>
26 </a>
27 </div>
28 <span id="main" onclick="openNav()">
29 &#9776;
30 </span>
31 <!-- end menu lateral -->

```

Figure 2.24 : capture 1 de code html pour bar d’option

```

30 </span>
31 <!-- end menu lateral -->
32 <!-- الجزء الفوقاني -->
33 <div style="height:60px; background: #00909E;" class="nav" >
34 <a class="nav-link active" href="#">Home</a>
35 <div data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
36 <a class="nav-link" href="#">Fichier</a></div>
37 <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
38 <a class="dropdown-item" href="#mu-contact">ouvrir</a>
39 <a class="dropdown-item" href="#"><span>nouveau </span></a>
40 <a class="dropdown-item" href="#"><span>enregistrer </span></a>
41 <a class="dropdown-item" href="#"><span>enregistrer sous ....</span></a></div>
42 <a class="nav-link" href="#">Quitter</a>
43 <a class="nav-link" href="#">Edition</a>
44 <a class="nav-link" href="#">Aide</a>
45 </div>
46 <!-- نهاية الجزء الفوقاني -->
47
48 <!-- الجزء الثاني -->
49 <div style="height: 100%; background: yellow; display: flex;">
50 <!-- القائمة التي بالنظر -->
51 <div style="width: 100px; display: flex; background: aqua;" class="btn-group-vertical" role="group" >
52 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
53 <p>home</p> </button>
54 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
55 <p>fichier</p></button>
56 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
57 <p>edition</p></button>
58 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
59  <p>aide</p> </button>
60 <button type="button" style="background-color: #142850;" class="btn btn-secondary">

```

Figure 2.25 : capture 2 de code html pour bar d'option

```

35 <div data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
36 <a class="nav-link" href="#">Fichier</a></div>
37 <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
38 <a class="dropdown-item" href="#mu-contact">ouvrir</a>
39 <a class="dropdown-item" href="#"><span>nouveau </span></a>
40 <a class="dropdown-item" href="#"><span>enregistrer </span></a>
41 <a class="dropdown-item" href="#"><span>enregistrer sous ....</span></a></div>
42 <a class="nav-link" href="#">Quitter</a>
43 <a class="nav-link" href="#">Edition</a>
44 <a class="nav-link" href="#">Aide</a>
45 </div>
46 <!-- نهاية الجزء الفوقاني -->
47
48 <!-- الجزء الثاني -->
49 <div style="height: 100%; background: yellow; display: flex;">
50 <!-- القائمة التي بالنظر -->
51 <div style="width: 100px; display: flex; background: aqua;" class="btn-group-vertical" role="group" >
52 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
53 <p>home</p> </button>
54 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
55 <p>fichier</p></button>
56 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
57 <p>edition</p></button>
58 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
59  <p>aide</p> </button>
60 <button type="button" style="background-color: #142850;" class="btn btn-secondary">
61 <p>quitter</p></button>
62 </div>
63 <!-- نهاية القائمة التي بالنظر -->
64

```

Figure 2.26: capture 3de code html pour bar d'option

- **Menu latéral :**

- ✓ **Programme html**

Le figure 2.27 présent le code html qui donne le menu latéral, ce menu présenté selon le class « sidenav » pour l’affichage de l’image de logo

Et un icone au-dessus pour revient à la page home, et span « main ».

Ce deux class couplé avec deux fonctions de JavaScript « closenav () et opennav () ».

Nous avons donné le caractère (z-index :1 ;) dans le fichier de css pour le class sidnav.


```

18 <!-- menue lateral -->
19 <div id="mysidenav" class="sidenav">
20 <a href="#" class="closebtn" onclick="closeNav()">
21 &times;
22 <br>
23 <br>
24 
25 </br>
26 </a>
27 </div>
28 <span id="main" onclick="openNav()">
29 &#9776;
30 </span>
31 <!-- end menu lateral -->

```

Figure 2.27 : capteur 1 de code html pour le menu latéral

✓ **Programme JavaScript**

Les deux fonctions de JavaScript « closeNav () et openNav () » montre dans le (figure 2.28) permet le class sidenav qui affiche quand a fait clique sue le sapan « main ».

```

1 function openNav(){
2   document.getElementById('mysidenav').style.width="250px";
3   document.getElementById('main').style.width="250px";
4 }
5 function closeNav(){
6   document.getElementById('mysidenav').style.width="0px";
7   document.getElementById('main').style.width="20px";
8 }

```

Figure 2.28: capteur 1 de code JavaScript pour le menu latéral

• **Programme html de différent interface :**

Les figures suivant montre le programme html et Vue.js qui lier entre les déférant interface dans l’application web, chaque interface c’est un fondamentalement par « section »qui définit au début par le code suivant :

< Section v-if = ‘ step == 1’>

Chaque fois définir le step par leur nombre de section.

✓ **Phase 1 :**

```

70 <section v-if="step == 1">
71 <div id="c3" class="card-deck">
72 <div style="background-color:#27496D;class="card text-white mb-3">
73 <div class="card-body">
74 <h3 class="card-title"><p>AVEC </p><p>BATTERIE</h3>
75 <h4 class="card-text">Dimensionnement d'un systeme photovoltaïque autonome avec batterie.</h4>
76 <br>
77 <button v-if="step==1"@click.prevent="nextstep" style="background-color:#00909E;class="btn btn-primary">
78   suivant</button> </a></p>
79 </div>
80 </div>
81 <div style="background-color:#27496D;class="card text-white mb-3">
82 <div class="card-body">
83 <h3 class="card-title"><p>RESEAU </p><p>RESEAU </h3>
84 <h4 class="card-text">Dimensionnement d'un systeme photovoltaïque connecte au reseau.</h4>
85 <br>
86 <div class="modal fade id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
87 <div class="modal-dialog">
88 <div class="modal-content">
89 <div id="d1" class="modal-header">
90 <h4 class="modal-title" id="exampleModalLabel">erreur</h4>
91 <button type="button" class="close" data-dismiss="modal" aria-label="Close">

```

Figure 2.29: capteur 1 de code html pour section1

```

96     <div class="modal-dialog">
97       <div class="modal-content">
98         <div id="d1" class="modal-header">
99           <h4 class="modal-title" id="exampleModalLabel">Erreur</h4>
100         <button type="button" class="close" data-dismiss="modal" aria-label="Close">
101           <span aria-hidden="true">&times;</span>
102         </button>
103       </div>
104       <div id="d1" class="modal-body">
105         Désolé, ce service n'est pas disponible
106       </div>
107       <div class="modal-footer">
108         <button type="button" class="btn btn-secondary" data-dismiss="modal">Sortir</button>
109       </div>
110     </div>
111   </div>
112 </div>
113
114 </div>
115 </div>
116
117 <div style="background-color:#274960; color:white; padding: 10px;">
118   <div class="card-body">
119     <hr>
120     <h3 class="card-title">POWPAGE </h3>
121     <hr>
122     <div class="card-text">Dimensionnement d'un système photovoltaïque sans batterie </small></p>
123     <!-- Button trigger modal -->
124     <button type="button" style="background-color:#009999; color:white; padding: 5px 15px; border: none;">
125       <span> </span>
126     </button>

```

Figure 2.30: capteur 2 de code html pour section1

```

119   <hr>
120   <h3 class="card-title">POWPAGE </h3>
121   <hr>
122   <div class="card-text">Dimensionnement d'un système photovoltaïque sans batterie </small></p>
123   <!-- Button trigger modal -->
124   <button type="button" style="background-color:#009999; color:white; padding: 5px 15px; border: none;">
125     <span> </span>
126   </button>
127   <!-- Modal -->
128   <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
129     <div class="modal-dialog">
130       <div class="modal-content">
131         <div id="d1" class="modal-header">
132           <h4 class="modal-title" id="exampleModalLabel">Erreur</h4>
133         <button type="button" class="close" data-dismiss="modal" aria-label="Close">
134           <span aria-hidden="true">&times;</span>
135         </button>
136       </div>
137       <div id="d1" class="modal-body">
138         Désolé, ce service n'est pas disponible
139       </div>
140       <div class="modal-footer">
141         <button type="button" class="btn btn-secondary" data-dismiss="modal">Sortir</button>
142       </div>
143     </div>
144   </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>

```

Figure 2.31: capteur 3 de code html pour section1

```

154     <hr>
155     <hr>
156     <h3 class="card-title">HYBRIDE </h3>
157     <hr>
158     <div class="card-text">Dimensionnement d'un système hybride.</h6>
159     <!-- Button trigger modal -->
160     <button type="button" style="background-color:#009999; color:white; padding: 5px 15px; border: none;">
161       <span> </span>
162     </button>
163     <!-- Modal -->
164     <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
165       <div class="modal-dialog">
166         <div class="modal-content">
167           <div id="d1" class="modal-header">
168             <h4 class="modal-title" id="exampleModalLabel">Erreur</h4>
169           <button type="button" class="close" data-dismiss="modal" aria-label="Close">
170             <span aria-hidden="true">&times;</span>
171           </button>
172         </div>
173         <div id="d1" class="modal-body">
174           Désolé, ce service n'est pas disponible
175         </div>
176         <div class="modal-footer">
177           <button type="button" class="btn btn-secondary" data-dismiss="modal">Sortir</button>
178         </div>
179       </div>
180     </div>
181   </section>
182   <section v-if="step == 2">
183     <table id="kln">
184       <thead>
185         <tr>
186           <th align="center">
187             <select v-model="location" @change="filllocationData" id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
188               <option selected>Choisissez un site</option>
189             </select>
190           </th>
191         </tr>
192         <tr>
193           <td align="center">
194             <input id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;" type="text" class="form-control mb-2" v-model="long">
195             <input id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;" type="text" class="form-control mb-2" v-model="lat">
196             <input id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;" type="text" class="form-control mb-2" v-model="alt">
197             <select id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
198               <option selected>date de début</option>
199               <option>2006/01/01</option>
200             </select>
201             <select id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
202               <option selected> date de fin </option>
203               <option>2006/12/31</option>
204             </select>
205           </td>
206         </tr>
207         <tr>
208           <td align="center">
209             <select id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
210               <option selected> site </option>
211               <option>mois</option>
212               <option>heure</option>
213               <option>année</option>
214               <option>jour</option>
215             </select>

```

Figure 2.32: capteur 4 de code html pour section1

✓ Phase 2 :

```

183   <section v-if="step == 2">
184     <table id="kln">
185       <thead>
186         <tr>
187           <th align="center">
188             <select v-model="location" @change="filllocationData" id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
189               <option selected>Choisissez un site</option>
190             </select>
191           </th>
192           <td align="center">
193             <input id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;" type="text" class="form-control mb-2" v-model="long">
194             <input id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;" type="text" class="form-control mb-2" v-model="lat">
195             <input id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;" type="text" class="form-control mb-2" v-model="alt">
196             <select id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
197               <option selected>date de début</option>
198               <option>2006/01/01</option>
199             </select>
200             <select id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
201               <option selected> date de fin </option>
202               <option>2006/12/31</option>
203             </select>
204           </td>
205         </tr>
206         <tr>
207           <td align="center">
208             <select id="zzz" style="background-color:#274960; color:white; padding: 5px 15px; border: none;">
209               <option selected> site </option>
210               <option>mois</option>
211               <option>heure</option>
212               <option>année</option>
213               <option>jour</option>
214             </select>

```

Figure 2.33: capteur 1 de code html pour section2

```
index.html | # javajs | # stylecss |
212 |
213 |
214 |
215 | <select id="tzz" style="background-color:#27496D;" class="form-control mb-2">
216 |   <option selected >Choisissez plan mode</option>
217 |   <option >plane horizontale</option>
218 |   <option >fixed titr@ fixe</option>
219 | </select>
220 |
221 | <select id="tzz" style="background-color:#27496D;" class="form-control mb-2">
222 |   <option selected > albedo </option>
223 |   <option >0,1</option>
224 |   <option >0,2</option>
225 | </select>
226 |
227 |
228 | <td align="center">
229 |   <div style="overflow:hidden;width: 600px;position: relative;"><iframe uidth="600" height="440"
230 |     src="https://maps.google.com/maps?width=600&map;height=440&hl=en&map;ouledy@icb(b11da)
231 |     &map;ie=UTF8&map;t-8&map;ioloc=8&map;output=embed" frameborder="0" scrolling="no" marginheight="0"
232 |     marginwidth="0"></iframe><div style="position: absolute;bottom: 10px;left: 0;right: 0;float: right;
233 |     auto;margin-right: auto;color: #000;text-align: center;"><small style="line-height: 1.8;font-size: 2px;background: #fff;">
234 |     Powered by <a href="https://embedgooglemaps.com/de/">embedgooglemaps DE</a> & <a href="https://mycasinonutansvenskilicens.net/">
235 |     Mycasinonutansvenskilicens.net</a>
236 |   </small></div></div><div style="background:none;position: absolute;bottom: 10px;float: right;
237 |     width: 100%;text-align: center;"></div></td>
238 | </tr>
239 | </table>
240 |
241 | <!-- End Google Map -->
242 | </section>
243 |
```

Figure 2.34: capteur 2 de code html pour section2

✓ Phase 3 :

```
index.html | # javajs | # stylecss |
243 | <section v-if="step == 3">
244 |   <table id="x" style="background-color:#27496D;"class="table table-hover table-dark" >
245 |     <thead>
246 |       <tr>
247 |         <th scope="col">composant</th>
248 |         <th scope="col">nombre</th>
249 |         <th scope="col">pulsissance</th>
250 |         <th scope="col">temps</th>
251 |       </tr>
252 |     </thead>
253 |     <tbody>
254 |       <tr>
255 |         <th scope="row">
256 |           <div class="form-row align-items-center">
257 |             <div class="col-auto">
258 |               <label class="sr-only" for="inlineFormInput">Name</label>
259 |               <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" id="inlineFormInput"
260 |                 placeholder="lamps" V-model="lamps">
261 |             </div>
262 |           </th>
263 |         </tr>
264 |       <tr>
265 |         <th scope="row">
266 |           <div class="form-row align-items-center">
267 |             <div class="col-auto">
268 |               <label class="sr-only" for="inlineFormInput">Name</label>
269 |               <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="A">
270 |             </div>
271 |           </th>
272 |         </tr>
273 |       <tr>
274 |         <th scope="row">
275 |           <div class="form-row align-items-center">
276 |             <div class="col-auto">
277 |               <label class="sr-only" for="inlineFormInput">Name</label>
278 |               <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="B">
279 |             </div>
280 |           </th>
281 |         </tr>
282 |       <tr>
283 |         <th scope="row">
284 |           <div class="form-row align-items-center">
285 |             <div class="col-auto">
286 |               <label class="sr-only" for="inlineFormInput">Name</label>
287 |               <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="C">
288 |             </div>
289 |           </th>
290 |         </tr>
291 |       <tr>
292 |         <th scope="row">
293 |           <div class="form-row align-items-center">
294 |             <div class="col-auto">
295 |               <label class="sr-only" for="inlineFormInput">Name</label>
296 |               <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" id="inlineFormInput"
297 |                 placeholder="television" V-model="tv">
298 |             </div>
299 |           </th>
300 |         </tr>
301 |       <tr>
302 |         <th scope="row">
303 |           <div class="form-row align-items-center">
304 |             <div class="col-auto">
305 |               <label class="sr-only" for="inlineFormInput">Name</label>
306 |               <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="D">
307 |             </div>
308 |           </th>
309 |         </tr>
310 |       </tbody>
311 |     </table>
312 |   </section>
313 |
```

Figure 2.35: capteur 1 de code html pour section 3

```
index.html | # javajs | # stylecss |
272 | </td>
273 | </tr>
274 | <tr>
275 |   <th scope="row">
276 |     <div class="form-row align-items-center">
277 |       <div class="col-auto">
278 |         <label class="sr-only" for="inlineFormInput">Name</label>
279 |         <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="B">
280 |       </div>
281 |     </th>
282 |   </tr>
283 |   <th scope="row">
284 |     <div class="form-row align-items-center">
285 |       <div class="col-auto">
286 |         <label class="sr-only" for="inlineFormInput">Name</label>
287 |         <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="C">
288 |       </div>
289 |     </th>
290 |   </tr>
291 |   <th scope="row">
292 |     <div class="form-row align-items-center">
293 |       <div class="col-auto">
294 |         <label class="sr-only" for="inlineFormInput">Name</label>
295 |         <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" id="inlineFormInput"
296 |           placeholder="television" V-model="tv">
297 |       </div>
298 |     </th>
299 |   </tr>
300 |   <th scope="row">
301 |     <div class="form-row align-items-center">
302 |       <div class="col-auto">
303 |         <label class="sr-only" for="inlineFormInput">Name</label>
304 |         <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="D">
305 |       </div>
306 |     </th>
307 |   </tr>
308 | </tbody>
309 | </table>
310 | </section>
311 |
```

Figure 2.36: capteur 2 de code html pour section 3

```

304 </td>
305 <td>
306 <div class="col-auto">
307 <label class="sr-only" for="inlineFormInput">Name</label>
308 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="E">
309 </div>
310 </td>
311 </tr>
312 </tbody>
313 <tbody>
314 <tr>
315 <td>
316 <div class="col-auto">
317 <label class="sr-only" for="inlineFormInput">Name</label>
318 <input id="1" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="F">
319 </div>
320 </td>
321 </tr>
322 <thead>
323 <tr>
324 <th colspan="2" scope="row">
325 <div class="form-row align-items-center">
326 <div class="col-auto">
327 <label class="sr-only" for="inlineFormInput">Name</label>
328 <input id="1" style="background-color:#27496D;" type="text" class="form-control mb-2" id="inlineFormInput" placeholder="ordi" V-model="pc">
329 </div>
330 </div>
331 </th>
332 <td>
333 <div class="col-auto">
334 <label class="sr-only" for="inlineFormInput">Name</label>

```

Figure 2.37: capteur 3 de code html pour section 3

```

331 </th>
332 <td>
333 <div class="col-auto">
334 <label class="sr-only" for="inlineFormInput">Name</label>
335 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="G">
336 </div>
337 </td>
338 </tr>
339 </tbody>
340 </table>
341 <div class="col-auto">
342 <label class="sr-only" for="inlineFormInput">Name</label>
343 <input id="1" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="H">
344 </div>
345 </td>
346 </tr>
347 </tbody>
348 </table>
349 <div class="col-auto">
350 <label class="sr-only" for="inlineFormInput">Name</label>
351 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="I">
352 </div>
353 </td>
354 </tr>
355 </tbody>
356 </table>
357 </tr>
358 <thead>
359 <tr>
360 <th colspan="2" scope="row">
361 <div class="form-row align-items-center">
362 <div class="col-auto">
363 <label class="sr-only" for="inlineFormInput">Name</label>
364 <input id="1" style="background-color:#27496D;" type="text" class="form-control mb-2" id="inlineFormInput" placeholder="refrigerateur" V-model="ref">
365 </div>
366 </div>
367 </th>
368 <td>
369 <div class="col-auto">
370 <label class="sr-only" for="inlineFormInput">Name</label>
371 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="K">
372 </div>
373 </td>
374 </tr>
375 </tbody>
376 </table>
377 <div class="col-auto">
378 <label class="sr-only" for="inlineFormInput">Name</label>
379 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="L">
380 </div>
381 </td>
382 </tr>
383 </tbody>
384 </table>
385 <div class="col-auto">
386 <label class="sr-only" for="inlineFormInput">Name</label>
387 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="M">

```

Figure 2.38: capteur 4 de code html pour section 3

```

357 </tr>
358 <thead>
359 <tr>
360 <th colspan="2" scope="row">
361 <div class="form-row align-items-center">
362 <div class="col-auto">
363 <label class="sr-only" for="inlineFormInput">Name</label>
364 <input id="1" style="background-color:#27496D;" type="text" class="form-control mb-2" id="inlineFormInput" placeholder="refrigerateur" V-model="ref">
365 </div>
366 </div>
367 </th>
368 <td>
369 <div class="col-auto">
370 <label class="sr-only" for="inlineFormInput">Name</label>
371 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="K">
372 </div>
373 </td>
374 </tr>
375 </tbody>
376 </table>
377 <div class="col-auto">
378 <label class="sr-only" for="inlineFormInput">Name</label>
379 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="L">
380 </div>
381 </td>
382 </tr>
383 </tbody>
384 </table>
385 <div class="col-auto">
386 <label class="sr-only" for="inlineFormInput">Name</label>
387 <input id="2" style="background-color:#27496D;" type="text" class="form-control mb-2" placeholder="123" V-model="M">

```

Figure 2.39: capteur 5 de code html pour section 3


```

380 </div>
381
382 </td>
383 <td>
384 <div class="col-auto">
385 <label class="sr-only" for="inlineFormInput"Name</label>
386 <input id="2" style="background-color:#274960;" type="text" class="form-control mb-2" placeholder="123" V-model="M">
387 </div>
388 </td>
389 </tr>
390 </tr>
391 <tr>
392 <tr valign="Middle">
393 <th scope=""></th>
394 <td>
395 <!-- Button trigger modal -->
396 </p>
397 </div>
398 </div>
399 <td>
400 </td>
401 <td>
402
403 </td>
404 </tr>
405 </tbody>
406 </table>
407 </h6>
408
409 </section>
410 <section v-if="step == 4">

```

Figure 2.40: capteur 6 de code html pour section 3

✓ Phase 4 :

```

409 <section v-if="step == 4">
410 <form id="M">
411 <div class="form-group">
412 <label for="exampleFormControlSelect1">Nombre autonome</label>
413 <select V-model="nombre_autonome" id="R" style="background-color:#274960;"class="form-control"
414 id="inputGroupSelect03" aria-label="Example select with button addon" name="grp2" required size="1">
415 <option style="color: #ffffff;" selected >...nombre autonome...</option>
416 <option style="color: #ffffff;" value="2">2</option>
417 <option style="color: #ffffff;" value="4">4</option>
418 </select>
419 <br>
420 </div>
421 <div class="form-group">
422 <label for="exampleFormControlSelect1">choix de panneau</label>
423 <select V-model="selecttwo" id="R" style="background-color:#274960;"class="form-control"
424 id="inputGroupSelect03" aria-label="Example select with button addon" name="grp2" required size="1">
425 <option style="color: #ffffff;" selected >...Choisissez un panneau...</option>
426 <option style="color: #ffffff;" value="280">panneau 1:puissance=280 watt</option>
427 <option style="color: #ffffff;" value="200">panneau 2 :puissance=200 watt</option>
428 </select>
429 <br>
430 </div>
431 <div class="form-group">
432 <label for="exampleFormControlSelect1">choix de batterie</label>
433 <select V-model="selectone" id="R" style="background-color:#274960;"class="form-control"
434 id="inputGroupSelect03" aria-label="Example select with button addon" name="grp2" required size="1">
435 <option style="color: #ffffff;" selected >choisissez une batterie...</option>
436 <option style="color: #ffffff;" value="2400">batterie 1:capacit =2000Ah et tension=12v</option>
437 <option style="color: #ffffff;" value="1200">batterie 2:capacit =1000Ah et tension=12v</option>
438 </select>
439 </div>

```

Figure 2.41: capteur 1 de code html pour section 4

Phase 5 :

```

441 </section>
442 <section v-if="step == 5">
443 <div class="section5">
444 <div id="B">
445 </div>
446 <div id="H">
447 </div>
448 <div id="L">
449 </div>
450 <div id="PP">
451 </div>
452 <div id="EE">
453 </div>
454 <div id="II">
455 </div>
456 <div id="TT">
457 </div>
458 </div>
459 <div class="col-8 offset-2 my-2">
460 <div class="card">
461 <div class="body-card">
462 <h3>graphe EPsyst</h3>
463 <hr>
464 <canvas id="myChart"></canvas>
465 </div>
466 </div>
467 </div>
468 </div>
469 </section>

```

Figure 2.42: capteur 1 de code html pour section

- **Code Vue.js de diff rant interface :**
La figure 2.43 montre le nombre premier et le nombre total des sections.

```

524     <script type="text/javascript">
525         const App = new Vue(
526             {
527                 el: "#app",
528                 data: {
529
530                     step: 1,
531                     totalstep: 5,

```

Figure 2.43: capteur 1 de code vue.js pour les déférant section
Le figure 2.44 montre la fonction qui permet affiche et cache les sections.

```

568         methods: {
569             prevstep: function() {
570                 this.step--;
571             },
572             nextstep: function() {
573                 this.step++;
574             },
575

```

Figure 2.44 : capteur 2de code vue.js pour les déférant section

- **Programme du bouton (Vue.js, html) :**

Chaque bouton dans l'application définie par (v-if) pour affiche dans leur section et coupler avec les fonctions (prevstep() , nextstep() ,getdatat(),datairr()).

```

476     <div style="height:70px;background: #142859;" >
477     <div align="center" >
478
479         <button v-if="step!=1" @click.prevent="prevstep" style="background-color: #000000;" class="btn btn-primary">
480             return</button>
481
482
483         <button v-if="step!=totalstep && step!=1" @click.prevent="nextstep" style="background-color: #000000;"
484             class="btn btn-primary" v-bind:disabled="location == ''>suivant</button>
485
486         <button v-if="step == 5" style="background-color: #000000;" class="btn btn-primary" v-on:click.prevent="getdata()" >
487             resultat</button>
488
489
490         <!-- Button trigger modal -->
491         <button v-if=" step == 2" type="button" style="background-color: #000000;"class="btn btn-primary"
492             data-toggle="modal" data-target="#exampleModal" v-on:click="datairr()">donner</button>
493
494     </div>
495
496 </div>
497
498 <!-- end button -->
499 </div>

```

Figure 2.45: capteur 1 de code de déférant bouton

2.5 Présentation de l'application :

Nous allons présenter dans ce qui suit les principales interfaces illustrant le fonctionnement de l'application :

- **Phase 1:** Le prototype réalisé s'identifie pour l'interface de l'étape 1 de l'application software « Figure 2.46 », l'ensemble des boutons est défini a droite dans l'interface pour identifier est choisir le type de système d'installation les autre boutons à gauche et dessous c'est un bar d'option pour faciliter le fonctionnement dans notre application , un foi l'utilisateur clique sur le bouton bleu « suivant » de choix «système autonome » l'application passé automatiquement à l'étape 2 , mais

que l'utilisateur clique sur les autres boutons « Figure 2.47 » l'application affiche un message d'erreur «Désolé, ce service n'est pas disponible». Le bouton « fichier » (figure 48) l'application affiche un box qui résume les quatre fonctionnements suivants :

Ouvrir : pour ouvrir un ancien projet de dimensionnement qui permet à l'utilisateur de continuer ou bien modifier dans leur projet.

Nouveau : permet l'utilisateur d'ouvrir un nouveau projet de dimensionnement
Enregistrer : permet l'utilisateur d'enregistrer leur travail dans le même projet.

Enregistre sous : permet l'utilisateur d'enregistrer leur travail dans n'importe quelle place dans leur pc de leur choix

Le bouton « aide » (figure 2.48) donne accès à une boîte qui donne la définition de notre application.

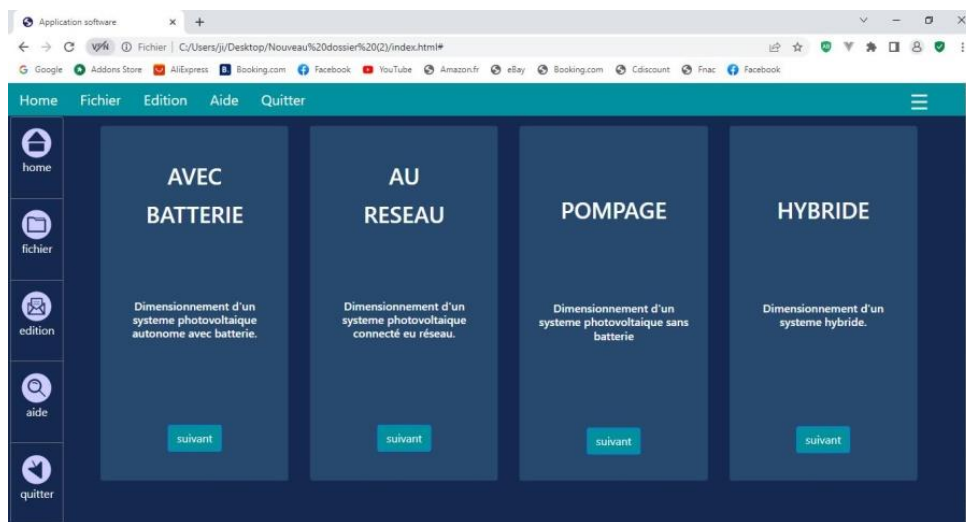


Figure 2.46 capture de 1^{ème} phase de l'application

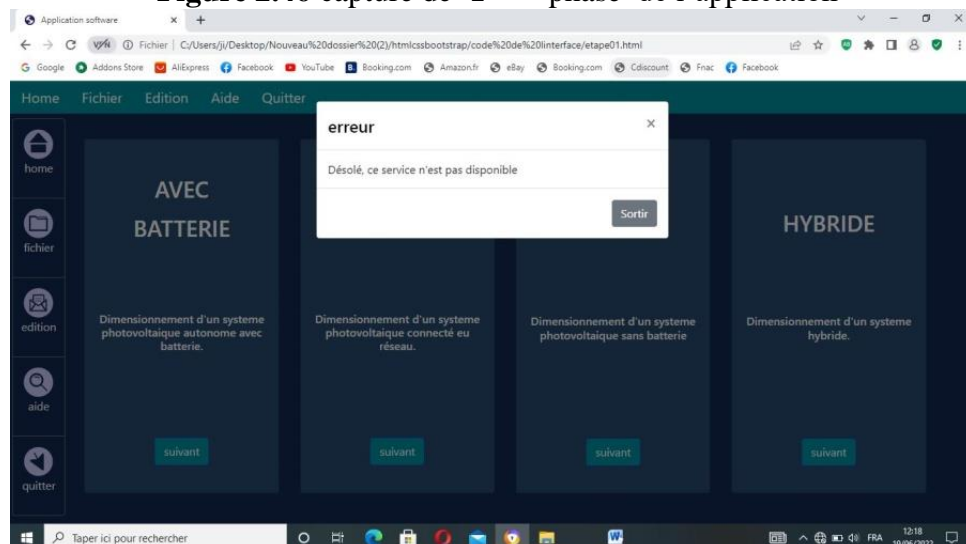


Figure 2.47 capteur de 1^{ème} phase exprimé le fonctionnement du bouton (suivant)

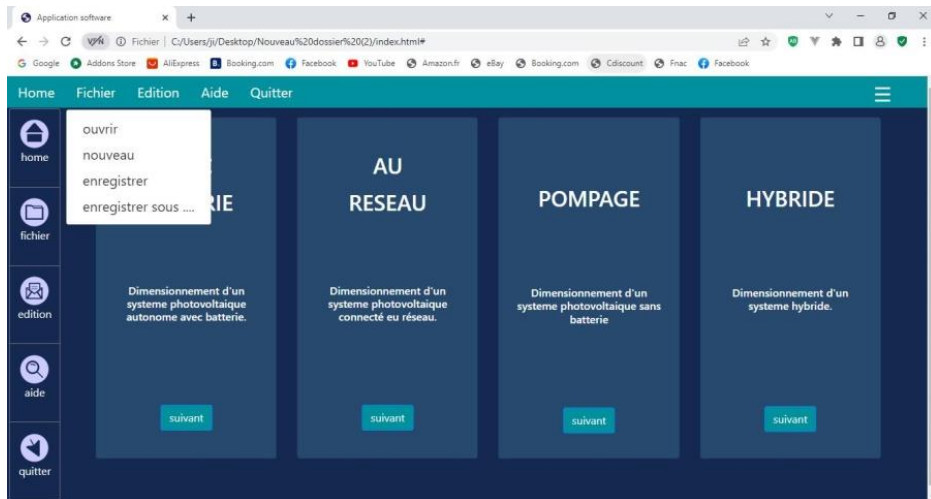


Figure 2.48 capture de 1^{ème} phase exprimée le fonctionnement du bouton (fichier)

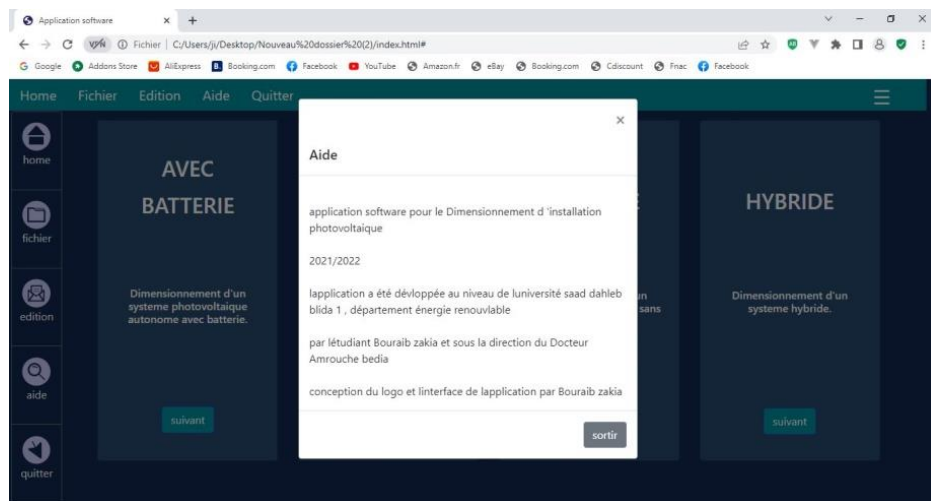


Figure 2.49 capture de 1^{ème} phase exprimé le fonctionnement du bouton (aide)

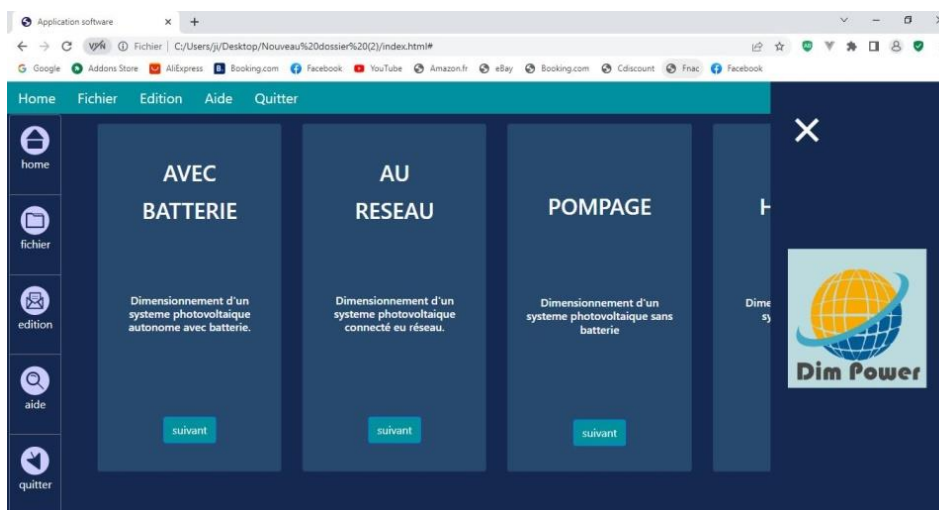


Figure 2.50 capture de 1^{ème} phase exprimer le menu latéral

- **Phase 3 :** La spécifique de cette phase (Figure 2.51) est la possibilité de choisir le site géographique à partir d'une carte géographique intégré, Les boutons à

gauche permet l'utilisateur choisir le site géographique l'application donner automatiquement leur caractéristique (longitude, latitude, altitude) puis l'utilisateur complété remplir les besoin (date de début, date de fin, inclinaison, time step, mode plans, albédo). Après avoir choisir le site et remplir leur caractéristique On clique sur le butons « donner » qui affiche un box qui contient leur irradiation

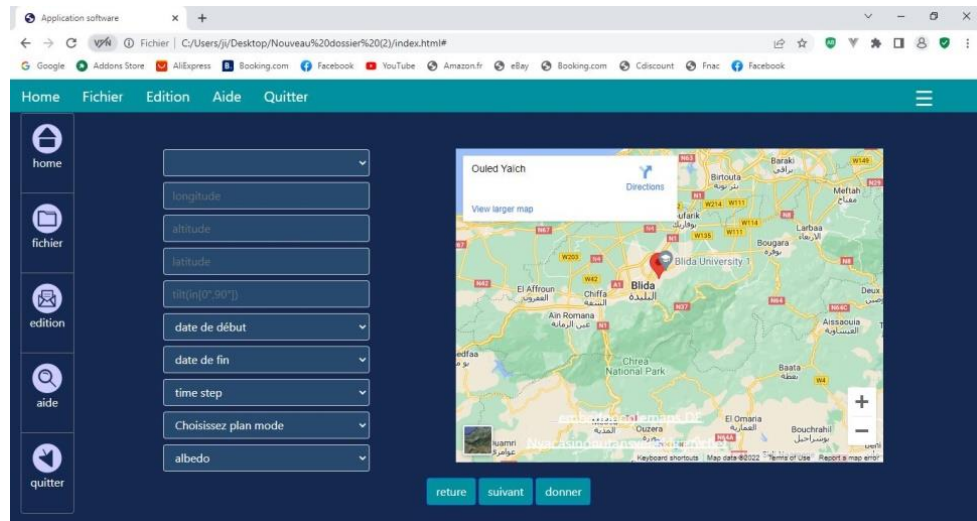


Figure 2.51 capture de 2^{ème} phase de l'application

- **Phase 4 :**

Cette phase nous donne un tableau qui permet à l'utilisateur de remplir leur besoin (nombre, temps, puissance), on clique sur le bouton « calculer » le logiciel va calculer l'énergie consommée et la puissance totale et donne aussi le type et la puissance de l'onduleur.

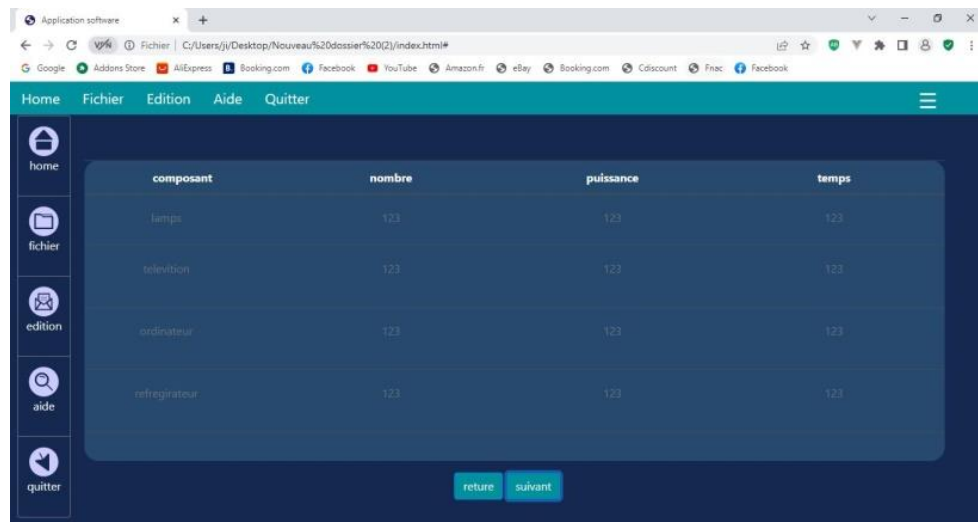


Figure 2.52: capture de 3^{ème} phase de l'application

- **Phase 5 :**

Cette phase (figure 2.53) introduit les choix des composantes « batteries, panneau » dans la spécification de l'application les choix peuvent se résumer en deux : le type « batterie, panneau » et leur caractéristique « capacité, tension, batterie » il existe deux choix dans chaque type de composant, ensuit

l'utilisateur clique sur le bouton calculer qui contient le nombre des composant nécessaire pour réaliser le dimensionnement

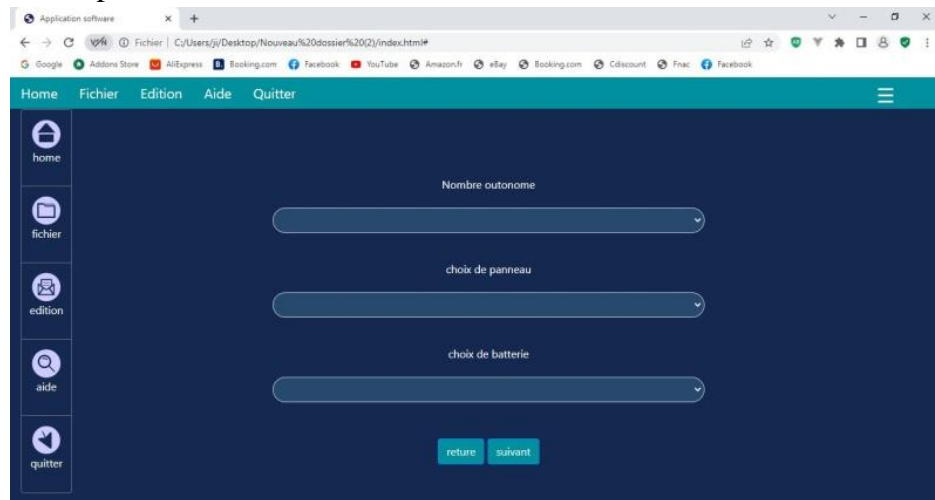


Figure 2.53 capture de 4 phase d'application

- **Phase 6 :**

Pour sauvegarder le résultat du système il faut appuyer sur le bouton « télécharger » le résultat sera stocké sous forme de fichier PDF, qui pourra être restauré pour Des prochains usages (figure 2.54).

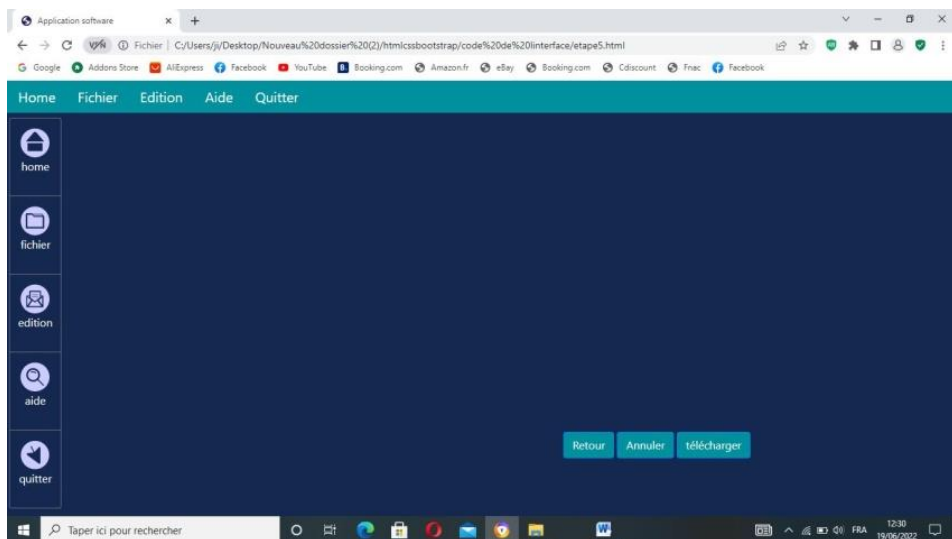


Figure 2.54 capture de 5^{ème} phase de l'application

2.6 Conclusion :

Durant ce chapitre nous avons présenté l'architecture conceptuelle de notre application en utilisant le formalisme UML, Nous avons défini les acteurs de notre application et les- tâches qu'ils assurent, puis élaboré le diagramme de cas d'utilisation, les diagrammes de séquence et le diagramme de classes, puis le diagramme de communication entre différent interface .

Ce chapitre a été consacré à la présentation de programme des diffèrent phases de notre application ainsi, nous avons présenté les donnée nécessaire de l'application.

Puis, nous sommes passés aux présentations de notre application en décrivant ses fonctionnalités et présentant plusieurs interfaces.

Dans le chapitre qui suit, nous allons faire notre développement de l'application software multiplateforme par JavaScript et vue.js et nous avons présentons le résultat finale de notre application à partir de logiciel électronjs.

Chapitre 3 :

Développement de l'application

3.1 Introduction :

JavaScript est l'un des langages de programmation les plus populaires au monde car il excelle non seulement dans la gestion des éléments sur le front-end ; Avec ce même langage, vous pouvez créer des applications mobiles multiplateformes, développer des API et travailler avec une infrastructure back end, et créer des applications de bureau,

Bien qu'il existe de nombreuses bibliothèques et Framework pour tous ces aspects, Electron se démarque lorsqu'il s'agit de développer des applications de bureau avec JavaScript,

Electron est un Framework JavaScript source ouvert permettant de créer des applications de bureau multiplateformes natives avec des technologies Web (HTML, CSS et JavaScript). Electron combine le moteur Chromium avec l'environnement d'exécution Node.js pour lire et exécuter votre base de code en tant que programme de bureau autonome. Et, pour prouver ses capacités, les programmes de bureau populaires créés à l'aide d'Electron incluent Slack, VS Code et Whatsapp.

Dans ce chapitre, nous allons utiliser Vue avec Electron pour développer notre application software.

3.2 Description du programme (Vue.js) :

3.2.1 programme des équations de dimensionnement :

3.2.1.1 tableaux de besoin et choix des équipements :

- **Les variables :** Pour permet l'utilisateur saisir et prendre des données dans notre application ces données doivent être enregistrées en tant que variables. La figure 3.1 montre les variables de la phase 4 et phase 5 (Figure 2.52 et Figure 2.53), ces variables sont définies dans l'élément « data » que nous précédemment défini dans chaque input de tableau et select de phase 4 avec la phrase « V-model » comme illustré dans le (Figure 3.1).

```

529     totalstep:5,
530     lamps:"",
531     tv:"",
532     pc:"",
533     ref:"",
534
535     A:"",
536     B:"",
537     C:"",
538
539     D:"",
540     E:"",
541     F:"",
542
543     G:"",
544     H:"",
545     I:"",
546
547     K:"",
548     L:"",
549     M:"",
550     selectone:[],
551     selecttwo:[],
552     nombre_autonome:[],
553     tilt:"",
554     lat:[],
555     long:[],
556     alt:[],
557     location:[],
558     },
559

```

Figure 3.1 : déclaration des variables de table de besoin et choix des panneaux
Et batteries et onduleur

- **Fonction vue.js** : La fonction `getdatat ()` illustrée dans la Figure 3.2 , contient tous les algorithmes d'équation nécessaires pour faire un dimensionnement d'un système photovoltaïque autonome, cette fonction calcule (énergie consommée, puissance utile et puissance de l'onduleur, nombre batterie et nombre panneaux) et donne également à l'utilisateur le type de l'onduleur nécessaire selon l'algorithme (if - else).

```

570     getdata:function(){
571         var EC=(this.A*this.B*this.C)+(this.D*this.E*this.F)+(this.G*this.H*this.I)+(this.K*this.L*this.M);
572         var PU =(this.A*this.B)+(this.D*this.E)+(this.G*this.H)+(this.K*this.L);
573         var PL=((PU/0.93)*1.25);
574         var nbatterie=(EC*2)/(this.selectone*0.8);
575         var nbpanneaux=((EC*2)/5)/(this.selecttwo*0.8);
576         if(PL<500){
577             document.getElementById("B").innerHTML="energie consomie="+EC ;
578             document.getElementById("H").innerHTML= "puissance utile" + PU ;
579             document.getElementById("L").innerHTML= "puissance de l'onduleur " + PL +"onduleur 12 v et " ;
580             document.getElementById("pp").innerHTML= "nombre batterie " + nbatterie ;
581             document.getElementById("ee").innerHTML= "nombre panneaux " + nbpanneaux;
582         }
583         else if(PL>500 && PL<2000 ){
584             document.getElementById("B").innerHTML="energie consomie="+ EC;
585             document.getElementById("H").innerHTML= "puissance utile" + PU ;
586             document.getElementById("L").innerHTML= "puissance de l'onduleur " + PL +"onduleur 24 v et " ;
587             document.getElementById("pp").innerHTML= "nombre batterie " + nbatterie ;
588             document.getElementById("ee").innerHTML= "nombre panneaux " + nbpanneaux;
589         }
590         else{
591             document.getElementById("B").innerHTML="energie consomie="+ EC ;
592             document.getElementById("H").innerHTML= "puissance utile" + PU ;
593             document.getElementById("L").innerHTML= "puissance de l'onduleur " + PL +"onduleur 48 v et " ;
594             document.getElementById("pp").innerHTML= "nombre batterie " + nbatterie ;
595             document.getElementById("ee").innerHTML= "nombre panneaux " + nbpanneaux;
596         }

```

Figure 3.2: fonction de vue.js pour calculer les besoin et choix des équipements

3.2.2 Irradiation annuelle :

3.2.2.1 choix de site : Lorsque l'utilisateur clique sur le choix de blida , les donnée géographique de leur site romplir automatiquement selon la fonction `filllocationdata`. (figure 3.3,figure 3.4,figure 3.5).

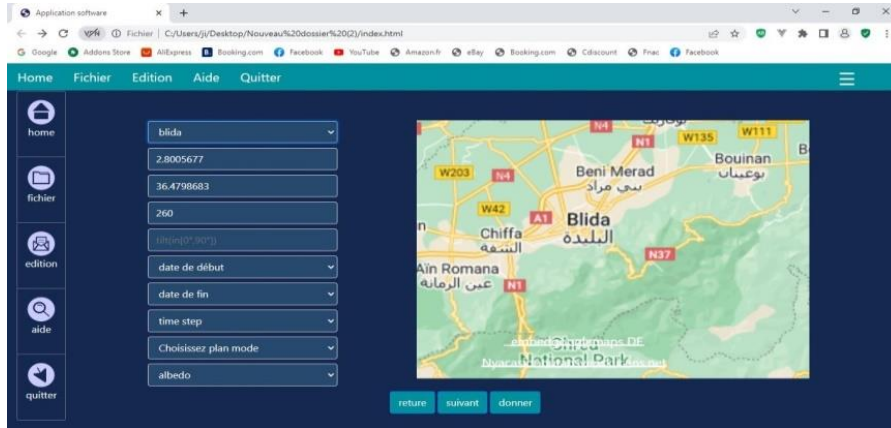


Figure 3.3 : phase 2 choix de site géographique

```

<td align="center">
  <select V-model="location" @change="filllocationData" id="zzz" style="background-color:#27496D;"
  class="form-control mb-2">
    <option selected>Choisissez un site</option>
    <option value="blida" >blida</option>
  </select>

```

Figure 3.4 : caractéristique vue.js pour choix de site géographique

```

642   filllocationData:function(){
643     var location={
644       blida:{lat:36.4798683,long:2.8005677,alt:260},
645     }
646     this.lat=location[this.location].lat;
647     this.long=location[this.location].long;
648     this.alt=location[this.location].alt;
649
650   },

```

Figure 3.5 : fonction vue.js pour choix de site géographique

3.2.2.2 les données d'irradiation : Les valeurs de rayonnement solaire que nous avons utilisées pour construire une base des données géographique de notre application software que nous avons téléchargé du site web (Solar radiation data « soda »), ces valeurs sont propres de la willaya Blida .figure 3.6 ,figure 3.7 et figure 3.8.

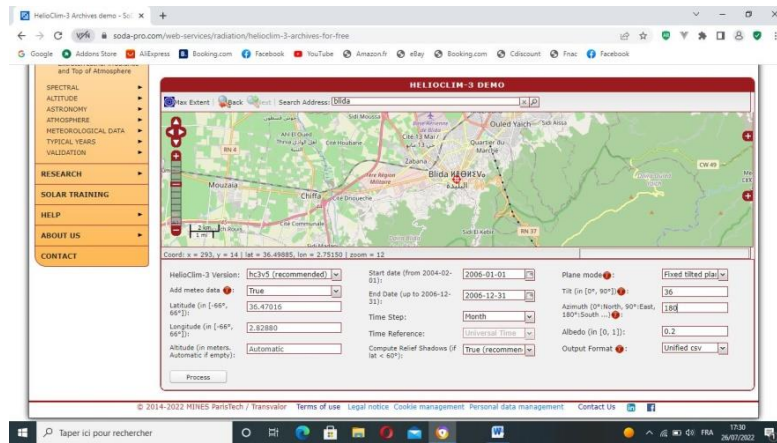


Figure 3.6 : interface de site web (Solar radiation data « soda »)

The screenshot shows an Excel spreadsheet titled 'irradiationblida - Microsoft Excel'. The data is organized in a table with columns for Date, Time, Direct Inclined, Diffuse Inclined, Reflected, Global Inclined, Direct Horiz, Diffuse Horiz, Global Horiz, Clear-Sky, Top of Atmosphere, and Nb valid. The data covers the period from 28/02/2006 to 31/12/2006. The 'Global Inclined' column is highlighted in yellow.

#	Date	Time	Direct Inclined	Diffuse Inclined	Reflected	Global Inclined	Direct Horiz	Diffuse Horiz	Global Horiz	Clear-Sky	Top of Atmosphere	Nb valid
37	31/01/2006	24:00:00	51815	2974	1840	85408	29402	28379	57976	105663	152243	31
38	28/02/2006	24:00:00	63504	13991	2353	99847	40683	33468	74151	127592	177547	28
39	31/03/2006	24:00:00	118407	45803	4427	168637	94244	45291	139534	187793	254526	31
40	30/04/2006	24:00:00	102326	52233	4822	159382	96407	55562	151968	209895	298626	31
41	31/05/2006	24:00:00	100390	58076	5504	163970	100950	64503	173453	236584	344433	31
42	30/06/2006	24:00:00	118060	53926	6267	178274	130618	60998	197514	238669	346732	31
43	31/07/2006	24:00:00	150766	51458	7395	215619	170279	56789	233068	246771	349853	31
44	31/08/2006	24:00:00	152790	50633	6551	209974	153408	53056	206464	227093	319843	31
45	30/09/2006	24:00:00	125854	46648	4833	177325	106225	46085	152310	185738	263416	29
46	31/10/2006	24:00:00	125144	43555	3984	172484	86510	39060	125560	148949	214588	31
47	30/11/2006	24:00:00	103058	34819	2813	140690	59321	29321	88641	107293	158063	31
48	31/12/2006	24:00:00	59784	30841	1878	92503	31100	28090	59190	92372	138600	31

Figure 3.7 : capteur 1 fichier Excel d'irradiation

The screenshot shows an Excel spreadsheet titled 'irradiationblida - Microsoft Excel'. The data is organized in a table with columns for Global Horiz, Clear-Sky, Top of Atmosphere, Nb valid days, Temperature, lative Humid, Pressure, Wind speed, Wind direction, Rainfall, Snowfall, and Snow depth. The data covers the period from 28/02/2006 to 31/12/2006. The 'Global Horiz' column is highlighted in yellow.

#	Global Horiz	Clear-Sky	Top of Atmosphere	Nb valid days	Temperature	lative Humid	Pressure	Wind speed	Wind direction	Rainfall	Snowfall	Snow depth
37	57976	105663	152243	31	281.10	78.88	972.69	0.32	212.71	86.289817	0.452255	0.000024
38	74151	127592	177547	28	281.91	77.28	970.07	1.72	254.61	81.083120	1.275520	0.000197
39	139534	187793	254526	31	285.94	71.22	970.49	2.40	261.86	19.099174	0.000000	0.000000
40	151968	209895	298626	30	285.86	65.19	968.28	0.26	218.10	17.858392	0.000000	0.000000
41	173453	236584	344433	31	293.85	66.77	970.31	0.67	43.19	148.553136	0.000000	0.000000
42	197514	238669	346732	30	296.39	52.21	971.25	0.64	105.05	36.423180	0.000000	0.000000
43	233068	246771	349853	31	300.51	48.09	971.99	1.28	33.26	6.051665	0.000000	0.000000
44	206464	227093	319843	31	299.07	52.47	969.29	1.51	15.59	3.203095	0.000000	0.000000
45	152310	185738	263416	29	296.35	56.17	970.53	0.79	296.74	18.654588	0.000000	0.000000
46	125560	148949	214588	31	294.32	54.76	971.05	0.83	232.61	17.211182	0.000000	0.000000
47	88641	107293	158063	30	288.90	65.25	975.18	1.43	204.22	35.912306	0.000000	0.000000
48	59190	92372	138600	31	284.15	76.66	978.10	1.49	267.34	131.432131	0.095489	0.000000

Figure 3.8 : capteur 2 fichier Excel d'irradiation

3.2.2.3 tables d'irradiation :

Le figure suivant (Figure 3.9) montre une table HTML, elle contient les valeurs de rayonnement solaire que nous avons précédemment téléchargées sur le site web (Solar irradiation data). Ce tableau est inclus dans le code de la fonction dataIrr () (Figure 3.9).

```

1
2 <table>
3 <tr> <td>Date</td><td> >Direct Inclined</td><td> >Diffuse Inclined</td><td>Global Inclined</td><td>Direct Horiz</td><td>Diffuse Horiz</td>
4 <td>Global Horiz</td></tr> <tr><td>Janv</td><td> >53819</td><td> >29749</td><td>>85408</td><td><td>29400</td><td>>28575</td><td>>579796</td>
5 <tr><td>Fevr</td><td> >63504</td><td> >33991</td><td>>99847</td><td><td>40683</td><td><td>33468</td><td><td>74151</td></tr>
6 <tr> <td>Mars</td><td> >118407</td><td> >45803</td><td>>168637</td><td><td>94244</td><td><td>45291</td><td><td>139534</td>
7 </tr><tr><td>Avr</td><td> >102326</td><td> >52233</td><td>>159382</td><td><td>96407</td><td><td>55562</td><td><td>151968</td></tr>
8 <tr><td>Mai</td><td> >100390</td><td> >58076</td><td>>163970</td><td><td>108950</td><td><td>64503</td><td><td>173453</td></tr>
9 <tr><td>Juin</td><td> >118080</td><td> >53926</td><td>>178274</td><td><td>136616</td><td><td>60898</td><td><td>197514</td></tr>
10 <tr><td>Juil</td><td> >156766</td><td> >51458</td><td>>215619</td><td><td>176279</td><td><td>56789</td><td><td>233068</td></tr>
11 <tr> <td>out</td><td> >152790</td><td> >50633</td><td>>209974</td><td><td>153408</td><td><td>53059</td><td><td>206464</td></tr>
12 <tr><td>Sept</td><td> >125854</td><td> >46648</td><td>>177335</td><td><td>106225</td><td><td>46085</td><td><td>152310</td></tr>
13 <tr><td>Octb</td><td> >125144</td><td> >43355</td><td>>172484</td><td><td>86510</td><td><td>39050</td><td><td>125560</td></tr>
14 <tr><td>Novb</td><td> >103028</td><td> >34819</td><td>>140690</td><td><td>59321</td><td><td>29321</td><td><td>88641</td></tr>
15 <tr> <td>Decm</td><td> >59784</td><td> >30841</td><td>>92503</td><td><td>31100</td><td><td>28090</td><td><td>59190</td></tr>
16 <tr></tr></table>;
17

```

Figure 3.9 : table html pour les donner de l’irradiation

3.2.2.3 fonctions d’affichage d’irradiation :

La fonction dataIrr () illustrée dans la Figure 3.10, notamment en montrant les valeurs de tableau de rayonnement solaire dans la phase 2 (Figure 2.51).

Si l’utilisateur ne sélectionne aucun emplacement géographique, l’application affichée un message d’erreur (veuillez sélectionner votre site géographique) –Figure 3.11-

Dans le cas du choix d’un emplacement géographique, une fenêtre apparait avec la valeur de rayonnement solaire pour l’emplacement – figure 3.12-

```

652 dataIrr:function(){
653   if(this.location!=""){
654     document.getElementById("bkh").innerHTML="<table><tr><td>Date</td><td> >Direct Inclined</td><td> >Diffuse Inclined</td><td>Global Incline
655   }
656   else{
657     document.getElementById("bkh").innerHTML="veuillez sélectionner votre site géographique";
658   }
659 },

```

Figure 3.10: fonction Vue.js pour l’affichage de l’irradiation

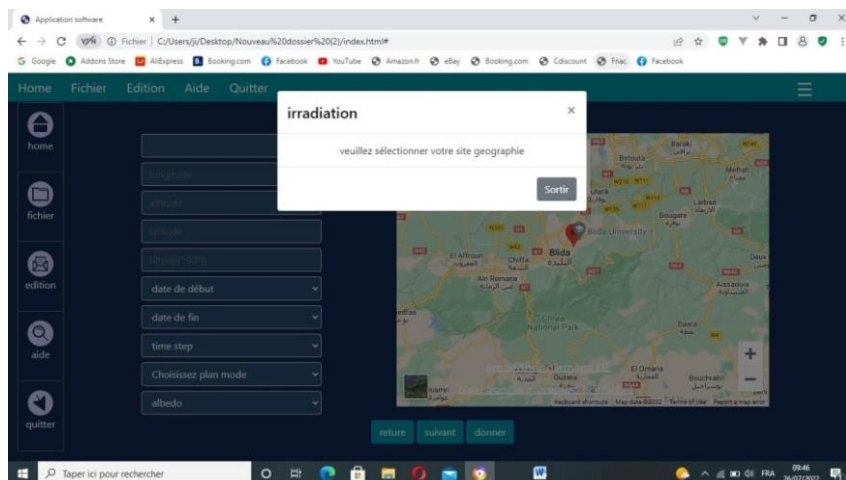


Figure 3.11: résultat 1 de La fonction dataIrr ()

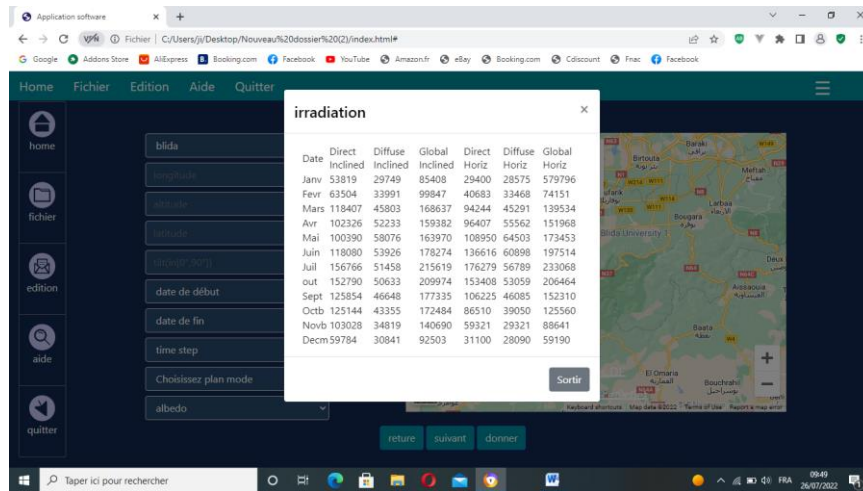


Figure 3.12: résultat 2 de La fonction dataIrr ()

3.2.3 Énergie produit

- Fonction de calculer et d'affiche l'énergie produit de système :**
 pour calculer la valeur l'énergie produit de système nous avons stocker les valeurs : (température moyenne de mois, indice de clarté , irradiation globale de site) chacun dans un tableau , puis utilisé trois variable dans lesquelles nous avons stocké trois tables vides pour les valeurs suivantes : (température de cellule , rendement moyenne, l'énergie produit par cellule) , ensuite nous avons programmé un boucle qui calculer les valeurs précédents et stocké dans ses propres tables. Tout cela est montré dans la (figure 3.13).

```

598   var ta=[8.1,8.9,12.94,16.86,20.85,23.39,27.51,26.07,23.35,21.32,15.9,11.15];
599   var kt=[0.8,0.7,0.6,0.6,0.6,0.4,0.6,0.6,0.6,0.7,0.7,0.8];
600   var g=[8540,9984,16863,15938,16397,17827,21561,20997,17733,17248,14069,9250];
601
602   var tc=[];
603   var nm=[];
604   var ep=[];
605   for(var i=0;i<ta.length;i++){
606     var equation=((219+ 832*kt[i])*((45-20)/800))+ta[i];
607
608     tc.push(equation.toFixed(2));
609
610     var equation2=0.0075*(tc[i]-0.25);
611     nm.push(equation2);
612
613     var equation3=(5*nm[i]*(1-0.05)*(1-0.12)*g[i])*0.85;
614     ep.push(equation3.toFixed(2));
615   }
616   document.getElementById("i1").innerHTML=tc;
617
618   document.getElementById("tt").innerHTML=ep;
619   ---

```

Figure 3.13: programme de calcule l'énergie produit d'un système pv

- Fonction d'affiche de graphe de l'énergie produit de système :**
 Pour afficher les valeurs de l'énergie produit à l'intérieur d'une déclaration, nous avons utilisé une bibliothèque (chart.js) à partir de JavaScript (figure 3.14).

```

620
621
622     var ctx = document.getElementById('myChart').getContext('2d');
623     var chart = new Chart(ctx, {
624         // The type of chart we want to create
625         type: 'bar',
626
627         // The data for our dataset
628         data: {
629             labels: ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'juillet', 'Aout', 'Septembre', 'Octobre', 'Novembre', 'Décembre'],
630             datasets: [{
631                 label: 'ENERGY PRODUIT DE SYSTÈME',
632                 backgroundColor: 'rgb(255, 99, 132)',
633                 borderColor: 'rgb(255, 99, 132)',
634                 data: ep
635             }]
636         },
637         // Configuration options go here
638         options: {}
639     });
640
641

```

Figure 3.14: programme d'affichage de graphe l'énergie produit d'un système pv

3.3 Transformation de programme par électrons :

3.3.1 Description de la méthode de transformation

Tout d'abord, nous avons téléchargé « nodejs et npm » (figure 3.15), puis nous avons créé un fichier sous le nom « électronsjs »(figure 3.16) et nous chargerons le paquet d'électron à l'intérieur en utilisant la commande suivante : (git clone <https://github.com/electron/electron-quick-start>). (Figure 3.17)

Ensuite, nous l'ouvrons de l'intérieur du PowerShell dans un programme VS code et nous avons utilisé plusieurs commandes dans le terminal de VS code pour Transfer notre application web ver une application software. (Figure 3.18)

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\ji> node -v
v16.17.0
PS C:\Users\ji> npm -v
8.15.0
PS C:\Users\ji>

```

Figure 3.15: téléchargement node et npm

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\ji> mkdir electronjs

Répertoire : C:\Users\ji

Mode                LastWriteTime         Length Name
----                -
d-----          22/08/2022   12:27         electronjs

```

Figure 3.16: création de fichier électronsjs

```

Windows PowerShell

PS C:\Users\ji\electronjs> ls
PS C:\Users\ji\electronjs> git clone https://github.com/electron/electron-quick-start
Cloning into 'electron-quick-start'...
remote: Enumerating objects: 1174, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
Receiving objects: 100% (1174/1174), 962.87 KiB | 52.00 KiB/s, done.
Resolving deltas: 0% (0/714)
Resolving deltas: 100% (714/714), done.
PS C:\Users\ji\electronjs> ls

Répertoire : C:\Users\ji\electronjs

Mode                LastWriteTime         Length Name
----                -
d-----          22/08/2022   12:30             electron-quick-start

PS C:\Users\ji\electronjs> cd electron-quick-start
PS C:\Users\ji\electronjs\electron-quick-start> ls

Répertoire : C:\Users\ji\electronjs\electron-quick-start

Mode                LastWriteTime         Length Name
----                -
d-----          22/08/2022   12:30             .github
-a-----          22/08/2022   12:30              14 .gitignore
-a-----          22/08/2022   12:30             690 index.html
-a-----          22/08/2022   12:30            6585 LICENSE.md
-a-----          22/08/2022   12:30            1441 main.js
-a-----          22/08/2022   12:30           57108 package-lock.json
-a-----          22/08/2022   12:30             457 package.json
-a-----          22/08/2022   12:30             445 preload.js
-a-----          22/08/2022   12:30            1967 README.md
-a-----          22/08/2022   12:30             299 renderer.js
-a-----          22/08/2022   12:30             81 styles.css

PS C:\Users\ji\electronjs\electron-quick-start>

```

Figure 3.17: téléchargement des fichier git dans le fichier électron js

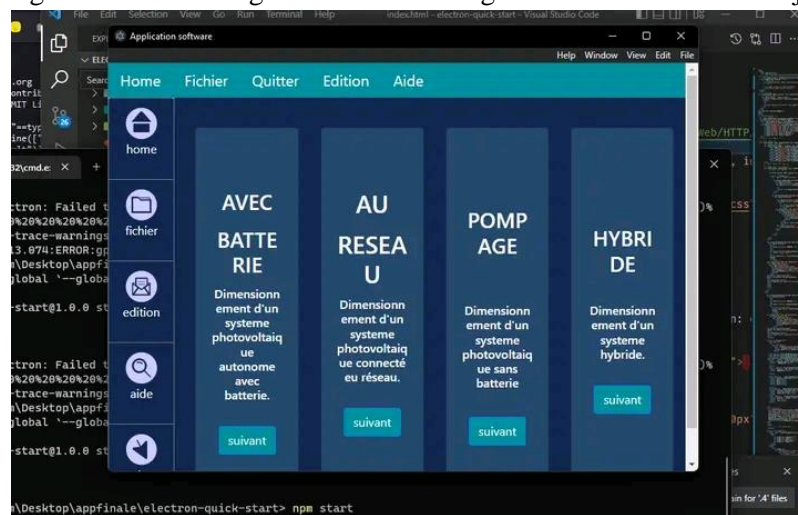


Figure 3.18 : capture Transformation de programme par électronjs

3.3.2 Présentation de résultat de transformation

Les figures suivant (figure 3.20 et figure 3.21, figure 3.22 , figure 3.23, figure 24 figure 25) représentent le résultat final de notre application software pour le dimensionnement d'un système photovoltaïque, après avoir traversé plusieurs étapes : (la conception, le développement et Transformation de programme par électronjs).

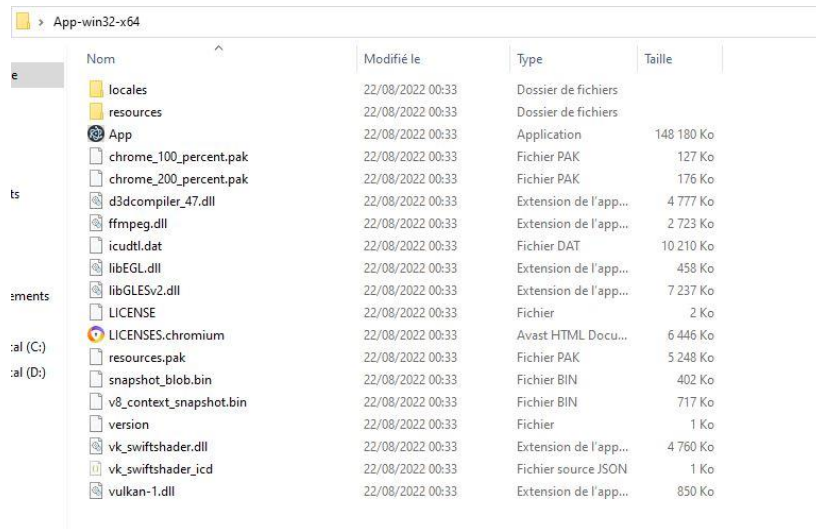


Figure 3.19 : les fichiers de notre application software

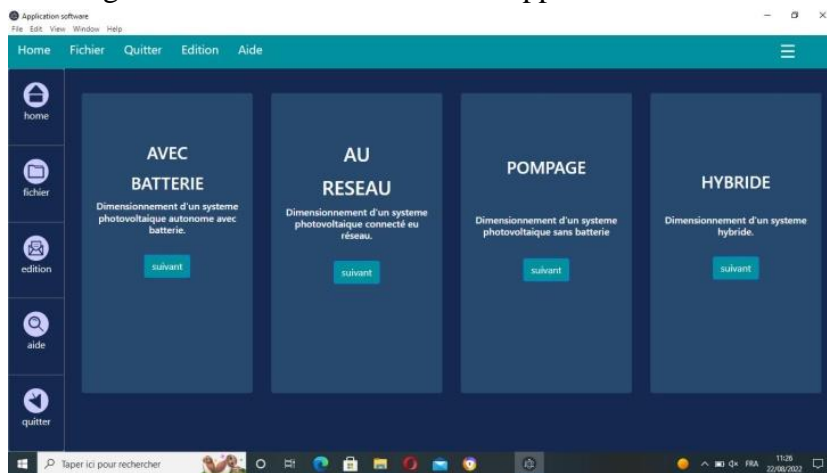


Figure 3.20 : phase 1 de notre application software

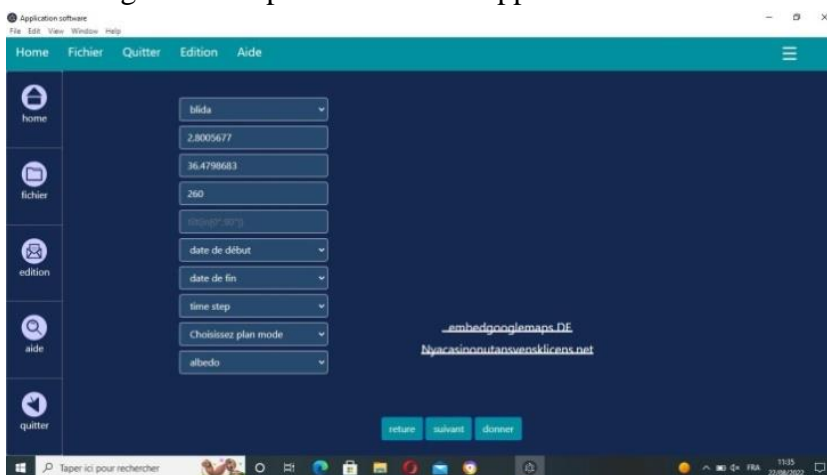


Figure 3.21 : phase 2 de notre application software

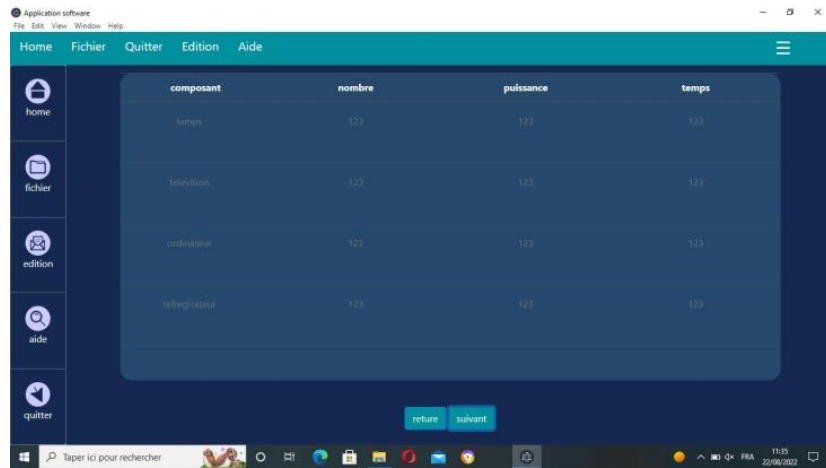


Figure 3.22 : phase 3 de notre application software

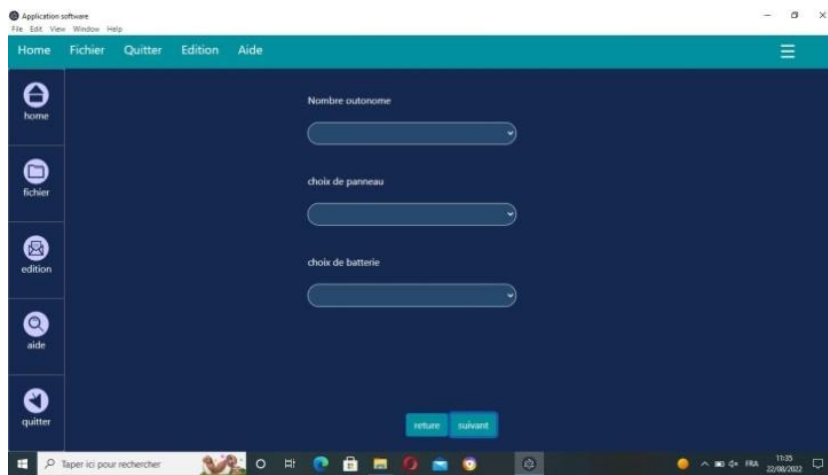


Figure 3.23 : phase 4 de notre application software

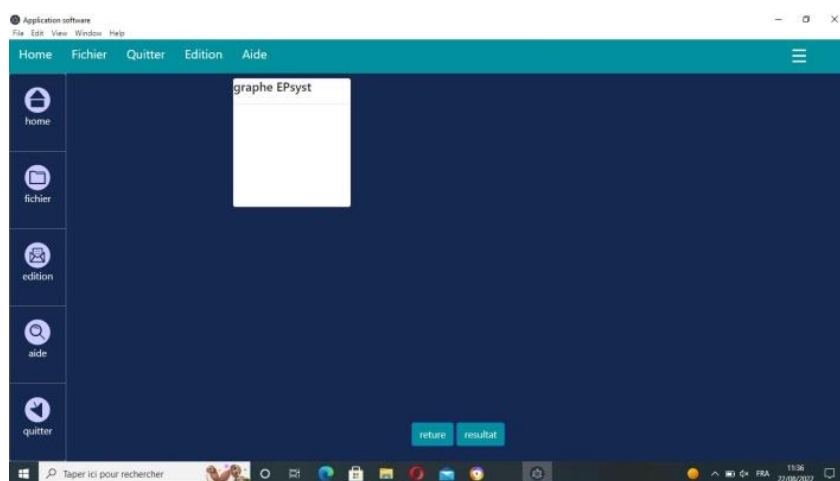


Figure 3.24 : phase 5 de notre application software

3.4 Conclusion :

Dans ce chapitre, nous avons développé notre application software, pour ce faire nous avons utilisé le langage de programmation Vue.js , et nous nous sommes appuyés sur plusieurs fonctions , dont celles-ci : La fonction getdatat () qui contient tous les algorithmes d'équation nécessaires pour faire un dimensionnement d'un système photovoltaïque autonome , et la fonction dataIrr () pour montrer les valeurs de rayonnement solaire , puis nous avons calculé les valeurs de l'énergie produit de système et nous avons montré dans un graphe par la bibliothèque (chart.js).

Ici, nous avons une application web que nous avons convertie à l'aide de la bibliothèque électronjs, enfin obtenir une application software multiplateforme pour le dimensionnement d'un système photovoltaïque.

Le chapitre qui suit, nous allons le consacrer à l'étude de cas utilisant notre application et autre programme (pvsyst) et nous comparerons le résultat.

Chapitre 4 :

Cas d'étude : dimensionnement d'une installation photovoltaïque

4.1 Introduction :

La production d'énergie est un défi de grande importance pour les années à venir. En effet, les besoins énergétiques des sociétés industrialisées ne cessent d'augmenter. Par ailleurs, les pays en voie de développement auront besoin de plus en plus d'énergie pour mener à bien leur développement. L'énergie solaire photovoltaïque provient de la transformation directe d'une partie du rayonnement solaire en énergie électrique. Cette conversion d'énergie s'effectue par le biais d'une cellule dite photovoltaïque (PV) basée sur un phénomène physique appelé effet photovoltaïque qui consiste à produire une force électromotrice lorsque la surface de cette cellule est exposée à la lumière. La tension générée peut varier en fonction du matériau utilisé pour la fabrication de la cellule. L'association de plusieurs cellules PV en série/parallèle donnent lieu à un générateur photovoltaïque (GPV) [15]. Figure 1.6 : logiciel Solar pro

Dans ce chapitre nous allons faire une étude d'une installation photovoltaïque autonome dans la willaya de Blida par notre application software et logiciel PVSYST. Cette étude détaillera les besoins en énergie du site et l'installation photovoltaïque nécessaire. Ensuite nous comparerons les deux résultats.

4.2 Dimensionnement par notre logiciel :

4.2.1 Données du site:

Au début, nous ouvrons notre application et choisissons le type de système (avec batterie) (figure 4.1 et figure 4.2).



Figure 4.1 : bouton de notre application(Dimpower)

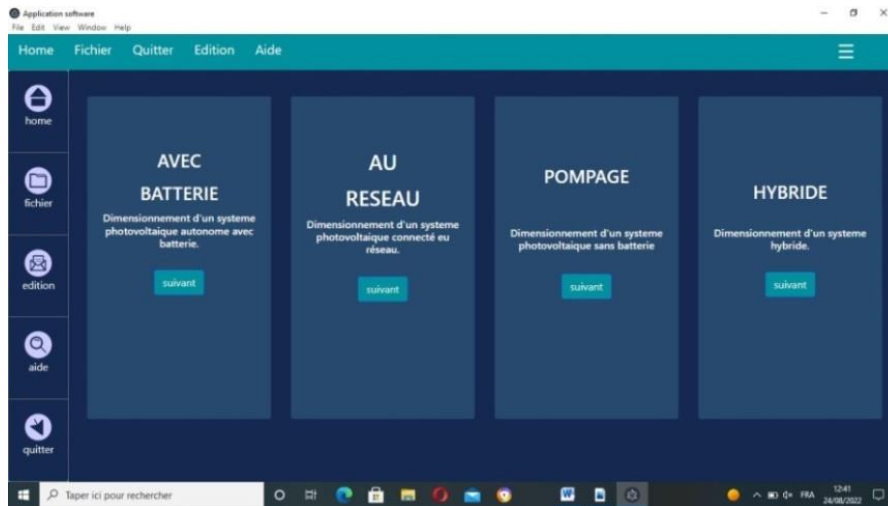


Figure 4.2 : 1^{er} interface de notre application(Dimpower)

Les coordonnées de la position de site Blida sont :

- Site géographique : Latitude 36 longitude 2.8 altitude 260 m.
- Angle d'inclinaison optimal: 36 degrés.
- Time step : mois.
- Choisissez plane mode : titre fixe.
- Albédo : 0.2. (figure 4.3)

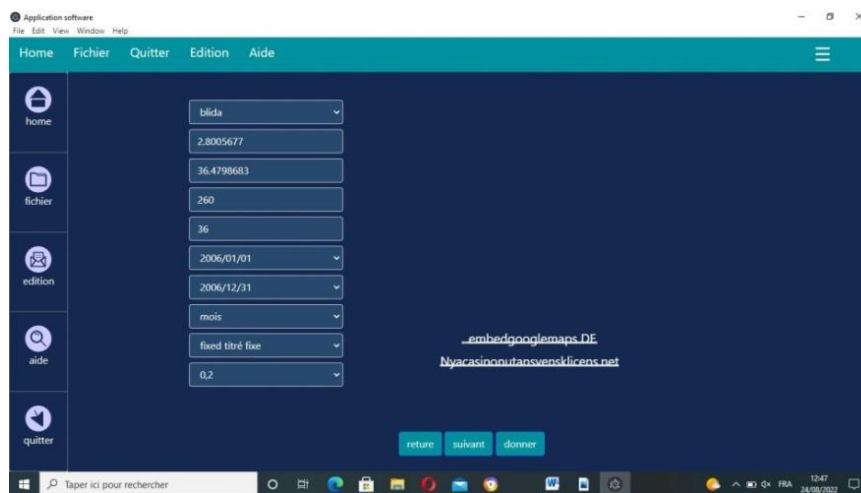


Figure 4.3 : choix de site géographique et autre caractéristique par notre application (Dimpower)

4.2.2 Les données météo du site :

Le figure suivant (figure 4.4) représente les données de l'éclairement du site de Blida durant l'année 01/01/2006 à 31/12/2006.

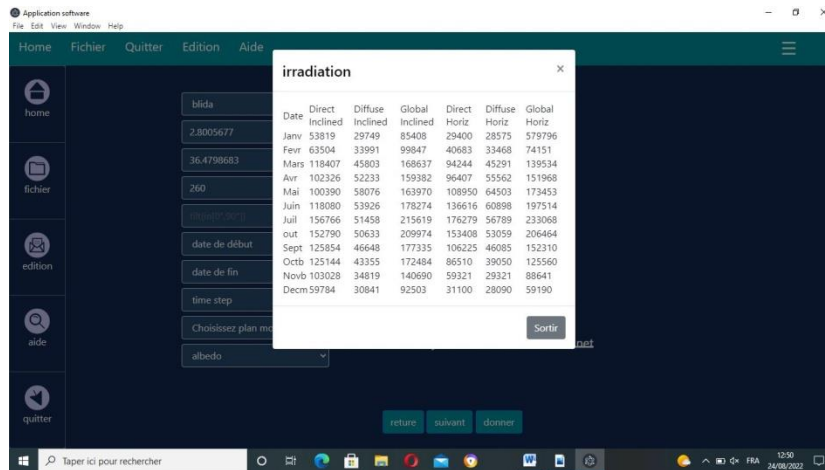


Figure 4.4 : table de l'irradiation

4.2.2 Besoin du site :

On suppose que notre installation alimente les appareils suivant (figure 4.5):

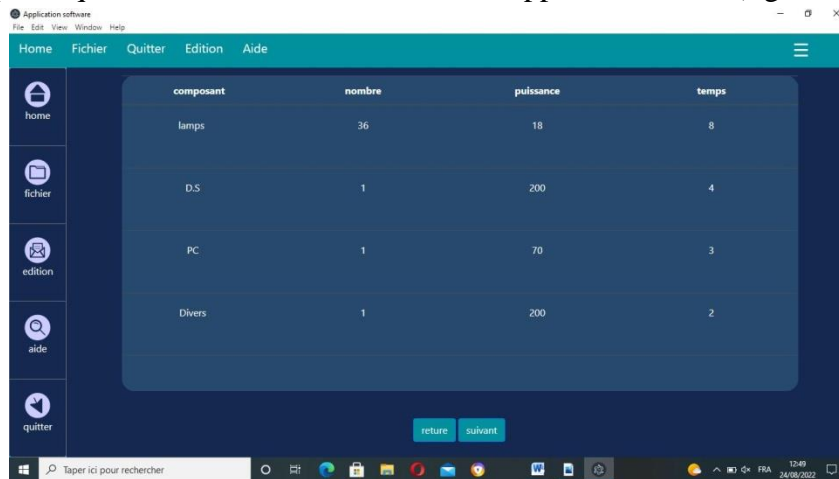


Figure 4.5 : table de besoin par notre application (Dimpower)

4.2.3 Etude du système :

La figure 4.6 représenté les choix des équipements de système :

- Nombre de jouer autonome : 2.
- Choix type de panneaux : puissance 280 W.
- Choix type de batterie : capacité 200Ah / tension 12 V.

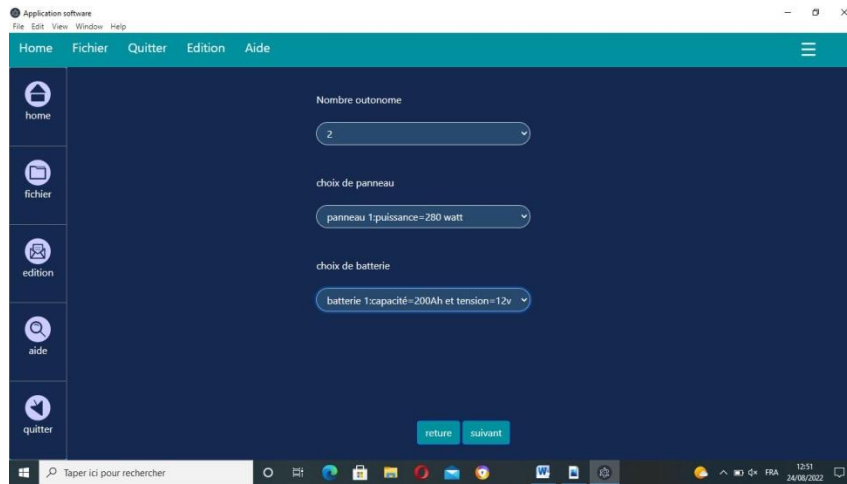


Figure 4.6 : les choix des équipements de système par notre application (Dimpower)

4.2.4 Résultats principaux :

La figure suivante (figure 4.7) représente le résultat final de notre simulation par notre application software. ce résultat contient deux partie : (graphe de l'énergie produit et paramètres essentiels de la simulation).

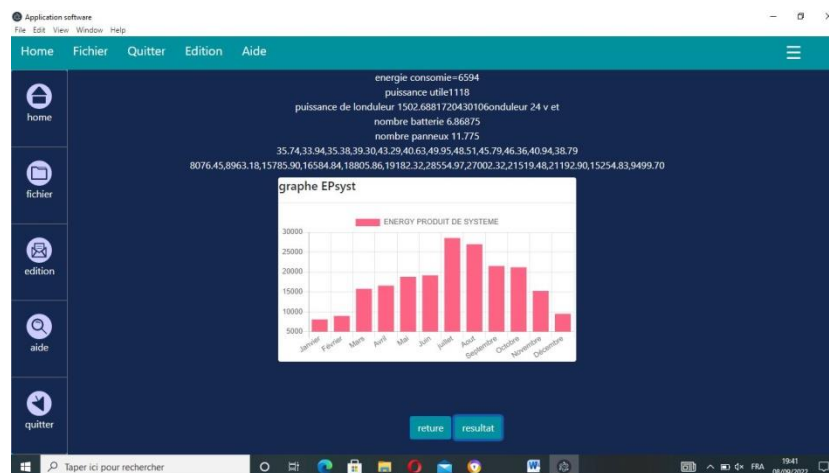


Figure 4.7 : le résultat final de notre simulation par notre application(Dimpower)

La figure (4.8) montre l'énergie produite par l'installation photovoltaïque donnée par notre application software durant une année. Sur cette figure on remarquera que l'énergie maximale est produite durant la période d'été (Juil –Aout).L'énergie minimale est produite durant le mois de janvier et décembre.

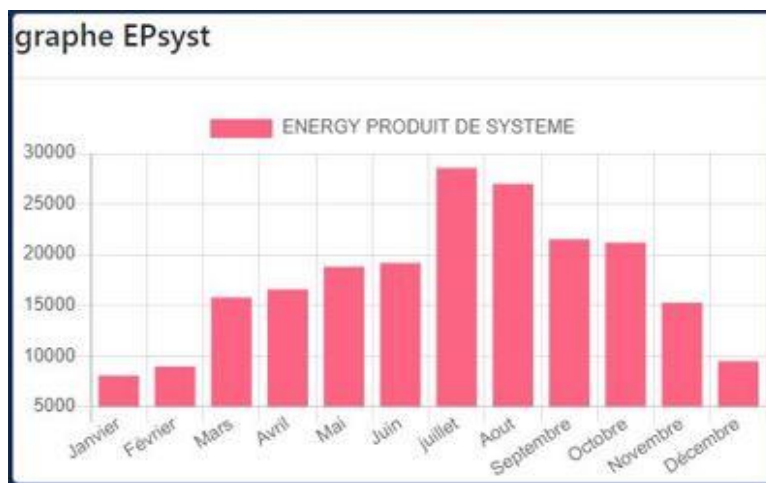


Figure 4.8 : lenergie produit de système photovoltaïque

Le tableau suivant (tableau 4.1) résume les paramètres essentiels de résultat de notre simulation utilisé dans notre application software pour le dimensionnement d'un système photovoltaïque.

Tableau 4.1 : résumer de résultat de simulation par notre application software

Résultat des équipements	Quantité
Energie consommé (wh)	6594
Puissance utile (w)	1118
Puissance de l'onduleur (w)	1502
Tension de l'onduleur (v)	24
Nombre batterie	6
Capacité batterie (Ah)	200
Tension batterie (v)	12
Nombre panneaux	12
Capacité panneaux (WC)	280

4.3 Dimensionnement par logiciel PVSYST :

4.3.1 Données du site:

La figure suivante représente Les coordonnées géographiques de la position de Blida.

Figure 4.9 : les coordonnées géographiques de site dan logiciel Pvsyst

4.3.2 Les données météo du site :

La figure (figure 4.10) représente les données de l'éclaircement du site de Blida.

	Irradiation globale horizontale kWh/m ² /mois	Irradiation diffuse horizontale kWh/m ² /mois	Température °C	Vitesse du vent m/s	Turbidité Linke []	Humidité relative %
Janvier	73.6	33.7	10.0	2.60	2.818	78.9
Février	87.1	38.6	10.6	2.80	3.078	75.3
Mars	129.4	57.7	13.4	2.90	3.939	72.0
Avril	156.8	80.7	15.7	3.00	4.385	71.3
Mai	185.8	91.6	19.4	3.11	5.062	66.2
Juin	212.3	85.6	23.8	3.19	5.251	59.9
Juillet	217.9	85.0	27.6	3.19	6.433	55.2
Août	198.3	80.7	27.8	3.00	5.486	56.8
Septembre	148.3	63.2	23.9	2.79	4.888	65.4
Octobre	117.4	49.7	20.6	2.40	4.021	66.4
Novembre	76.2	37.2	14.5	2.50	3.365	74.3
Décembre	67.6	31.5	11.3	2.39	2.922	78.2
Année	1670.7	735.1	18.2	2.8	4.304	68.3

Irradiation globale horizontale variabilité d'une année sur l'autre : 4.4%

Figure 4.10 : les données de l'éclaircement du site de Blida dans Pvsyst

4.3.2 Besoin du site :

Le figure suivant repris le tableau de besoin d'utilisateur par pvsyst

Nombre	Appareil	Puissance	Util. journ.	Distrib. horaire	Daily energy		
36	Lampes (LED ou fluo)	18 W/lampe	8.0 h/jour	OK	5184 Wh		
1	D.s	200 W/app	4.0 h/jour	OK	800 Wh		
1	pc	70 W/app	3.0 h/jour	OK	210 Wh		
0	Frigo / Congélateur	0.00 kWh/jour	24.0		0 Wh		
0	Machines à laver linge/vaisselle	0.0 W moy.	2.0 h/jour		0 Wh		
1	divers	200 W/app	2.0 h/jour	OK	400 Wh		
0	Autres utilisations	0 W/app	0.0 h/jour		0 Wh		
Consomm. de veille					1 W tot	24 h/jour	24 Wh
Energie journalière totale						6618 Wh/jour	
Energie mensuelle						198.5 kWh/mois	

Figure 4.11 : table de besoin par logiciel Pvsyst

4.3.3 Etude du système :

La figure 4.12 et figure 4.13 représenté les choix des équipement de système à partir de logiciel pvsyst:

- Choix type de panneaux : puissance 280 w.
- Choix type de batterie : capacité 200Ah / tension 12 v.

Définissez le pack de batteries	
Trier les batteries selon	<input checked="" type="radio"/> tension <input type="radio"/> capacité <input type="radio"/> fabricant
Tous les fabricants	12 V 200 Ah Pb Sealed Gel MPG 12V 200 Narada
Pb-acide	
2 batteries en série	Nombre de batteries: 6
3 batteries en parallèle	Nombre d'éléments: 36
100.0 % Etat d'usure initial (nb. de cycles)	Tension du pack batteries: 24 V
100.0 % Etat d'usure initial (statique)	Capacité globale: 600 Ah
	Energie stockée (80 % DOD): 11.5 kWh
	Poids total: 402 kg
	Nbre de cycles à 80 % DOD: 1000
	Energie totale stockée durant la vie de la batterie: 13176 kWh

Figure 4.12 : le choix de type de batterie par Pvsyst

Caractéristiques du champ de capteurs			
Module PV		Batterie	
Fabricant	Generic	Fabricant	Generic
Modèle	TWY280P672	Modèle	MPG 12V 200
(Base de données Pvsyst originale)		Technologie	Pb-acide, scellée, Gel
Puissance unitaire	280 Wc	Nombre d'unités	3 en parallèle x 2 en série
Nombre de modules PV	12 unités	Décharge : min. SOC	20.0 %
Nominale (STC)	3360 Wc	Energie stockée	11.9 kWh
Modules	12 Chaines x 1 En série	Caractéristiques du banc de batteries	
Aux cond. de fonct. (50°C)		Tension	24 V
Pmpp	3000 Wc	Capacité nominale	600 Ah (C10)
U mpp	32 V	Température	Fixée 10 °C
I mpp	94 A		
Régulateur		Seuils de régulation batterie	
Régulateur universel		Seuils de commande selon	Calcul SOC
Technologie	Convertisseur MPPT	Charge	SOC = 0.90 / 0.75
Coeff. de temp.	-5.0 mV/°C/Élem.	environ	27.4 / 25.6 V
Convertisseur		Décharge	SOC = 0.20 / 0.45
Efficacité maxi et EURO	97.0 / 95.0 %	environ	23.9 / 24.9 V
Puissance PV totale			
Nominale (STC)	3 kWc		
Total	12 modules		
Surface modules	23.3 m ²		
Surface cellule	21.0 m ²		

Figure 4.15 : le résultat final de notre simulation par Pvsyst

Le tableau suivant (tableau 4.2) résume les paramètres essentiels de résultat de notre simulation utilisé dans logiciel Pvsyst pour le dimensionnement d'un système photovoltaïque.

Tableau 4.2 : résumer de résultat de simulation par logiciel pvsyst

Résultat des équipements	Quantité
Energie consommé (Wh)	6618
Nombre batterie	6
Capacité batterie (Ah)	200
Tension batterie (v)	12
Nombre panneaux	12
Capacité panneaux (WC)	280

4.4 Comparaison entre les deux résultats :

D'après les deux résultats, le dimensionnement par les deux logiciel (Pvsyst et Dimpower) donner le même résultat, les équipements nécessaire pour faire cette installation photovoltaïque 6 batterie d'une capacité (200Ah et 12 v) avec 12 panneaux d'une capacité (280 WC). Nous remarquera que l'énergie maximale est produite durant la période d'été (Juin –Aout).L'énergie minimale est produite durant le mois de janvier et décembre.

4.4 Conclusion :

L'étude de ce chapitre a confirmé les résultats de notre application software (Dimpower) Pour le prouver, nous nous sommes appuyés sur une étude analytique par logiciel Pvsyst et nous avons comparé les deux résultats.

Le résultat de dimensionnement par notre logiciel et par logiciel Pvsyst sont identique.

Ainsi, notre application software (Dimpower) peut être adoptée comme une application pour le dimensionnement d'une installation d'un système photovoltaïque autonome.

CONCLUSION GENERALE

Le travail présenté dans ce mémoire concerne une application de dimensionnement des installations PV autonomes conformément au cahier de charge présenté au chapitre 1.

La première phase de notre travail a été de passer en revue les différentes étapes de la vie d'un projet photovoltaïque tout en décrivant le problème qui nous a poussé à concevoir la présente application. Ensuite nous avons présenté les différents outils de dimensionnement des installations solaires PV existants et ce, dans le premier chapitre.

Dans le deuxième volet nous sommes passés à la conception de l'application de dimensionnement solaire photovoltaïque. La présentation de ses fonctionnalités, les étapes du codage de conception, la description de l'organisation des différentes interfaces. Nous avons présenté la méthode de « UML », qui permettra de bien représenter les aspects fonctionnels statiques et dynamiques de notre projet par la série des trois diagrammes et le diagramme de communication qu'il offre, suivi d'une phase « conception de l'interface » s'appuyant sur la partie précédente de « UML ». La conception s'est faite en deux étapes : Maquettage avec Adobe Illustrator et Réalisation de programme selon la méthode SPA, Dans le deuxième chapitre.

Dans le troisième chapitre, nous avons présenté les détails de développement de l'application software, pour ce faire nous avons utilisé le langage de programmation Vue.js, et nous nous sommes appuyés sur plusieurs fonctions, dont celles-ci : La fonction `getdata()` qui contient tous les algorithmes d'équation nécessaires pour faire un dimensionnement d'un système photovoltaïque autonome, et la fonction `dataIrr()` pour montrer les valeurs de rayonnement solaire, puis nous avons calculé les valeurs de l'énergie produite de système et nous avons montré dans un graphe par la bibliothèque (`chart.js`).

Ici, nous avons une application web que nous avons convertie à l'aide de la bibliothèque `electronjs`, enfin obtenir une application software multiplateforme pour le dimensionnement d'un système photovoltaïque. Dans le dernier chapitre nous, avons effectué les tests de l'application en relevant les insuffisances de cette dernière. Ce chapitre a permis entre autre de faire deux études de cas. Le résultat de dimensionnement par notre logiciel et par logiciel `Pvsyst` sont identiques. Notons également ici que le nom proposé pour l'application dans ce mémoire est : "Dimpower" pour dire dimensionnement d'une puissance (power en anglais). Toutefois ce nom se verra changer lorsque l'application sera mise à disposition de tous les utilisateurs, mais cette version gardera son nom. Il est à souligner également que l'application fonctionne sur tous les systèmes d'exploitation, et d'après l'étude de la quatrième chapitre notre application software (Dimpower) peut être adoptée comme une application pour le dimensionnement d'une installation d'un système photovoltaïque autonome

Références bibliographiques :

- [1] C. Alonso, 'Contribution à l'Optimisation, la Gestion et le Traitement de l'Energie', Mémoire pour l'Habilitation à Diriger les Recherches, Université Paul Sabatier, Toulouse, 2003.
- [2] B. Multon, 'L'Energie Electrique: Analyse des Ressources et de la Production, Place des Sources Renouvelables', Journées Electrotechnique du Club EEA, 1999.
- [3] Ademe -Ines, 'Formation sur le Dimensionnement des Systèmes PV', (<http://www.ademe.fr/formation>)
- [4] L. Chancelier et E. Laurent, 'L'Electricité Photovoltaïque', Collection 'le Point sur', GRET, Ademe, Ministère de la coopération, 1995.
- [5]. R. Tchuidjan et M.J. Tchana Nkouimi, Outil d'aide au dimensionnement Des systèmes photovoltaïques domestiques, Université de Yaoundé I, Ecole Nationale Supérieure Polytechnique, le 25 Septembre 2009
- [6].énergie solaire photovoltaïque, dictionnaire environnement, www.actuenvirenement.com
- [7].composant et fonctionnement d'un système photovoltaïque, www.xpair.com
- [8].S.Semaoui et A.Hadjarab, I.hadj Mohamed, Serine, Y .Bakelli, Système photovoltaïque autonome de l'uraer Ghardaïa, centre de développement des énergies renouvelables, Algérie 2019
- [9] S. Makhloofi, "Contribution à l'optimisation des installations photovoltaïques par des Commandes intelligentes," Thèse de doctorat, Université de Batna 2, Algérie, 2013
- [10]. Chapitre 1 - Introduction to single page application, web Age solution, canada
- [11].single-page-application page avec définitions et exemple
- [12].Pascal Roques, UML2 Modéliser une application web, le cahier du programmeur, 4^{eme} édition, édition EYROLLES, paris, 2006
- [13].Verlain Foundikou, Méthodes de conception Orientés Objet, MCOO cours, 2012
- [14].C. Johnen, les concepts objets, UML Diagramme de classes, IUT de bordeaux, V2
- [15].Ameur Ali Elhabib , Daoudi Mustafa , Dimensionnement d'une installation photovoltaïque (application entreprise ENAVA) , Université Dr .Tahar Moulay de Saïda .