

Table des matières

Introduction.....	3
Partie1 : Conception et modélisation	3
1- Diagramme de cas d'utilisation.....	4
2 – Diagramme de séquence.....	6
2 -1 Ajouter un client	6
2 -2Authentification client	7
2 -3 Ajouter un Heart Beat.....	8
2 – Diagramme de class.....	9
Partie2 : Contexte, Technologies et Architecture	10
Rappel du contexte général de projet.....	11
1-Technologie et outils utilisés.....	12
2-Architecture	13
Partie3 : Résultats.....	14
1-interface admin	
2- interface client	
 Conclusion.....	 10
Références	11

Introduction

Le projet consiste à développer une application qui communique avec une montre connectée pour afficher les graphes de battements de cœur. Notre application contient deux interfaces : une interface administrateur et client. L'administrateur est concerné par l'ajout de clients et de la gestion des données de battements de cœur. Le client a le droit de consulter ses propres données de battements de cœur. Le développement de l'application s'est concentré sur la création d'une interface frontend conviviale qui communique avec des API pour effectuer des opérations telles que l'ajout de clients, la suppression de clients, la consultation des données de battements de cœur....

Dans ce rapport, nous aborderons différents aspects du projet, notamment la conception et la modélisation, le contexte du projet, l'architecture utilisée et les outils technologiques déployés. Nous concluons également avec un aperçu des résultats obtenus.

Partie 1: Conception et modélisation du projet

1- Diagramme de cas d'utilisation

Interface Admin :

- Admin peut s'authentifier
- Admin peut accéder au Dashboard
- Admin peut Consulter tous les clients
- admin peut faire des modifications sur tous les clients
- Admin peut ajouter un nouveau client
- Admin peut supprimer des clients déjà existant
- Admin consulter Heart Beat de tous les clients
- Admin consulter Heart Beat de tous les clients
- Admin peut ajouter des heart beat
- Admin peut rechercher des clients

Interface Client :

- Un client est identifié par son adresse mac, son numéro de téléphone, son email, nom, prénom et adresse
- Un client peut s'authentifier avec son adresse mac
- Un client possède un Heart beat
- Un Heart Beat possède 4 data
- Un client peut Consulter ses informations

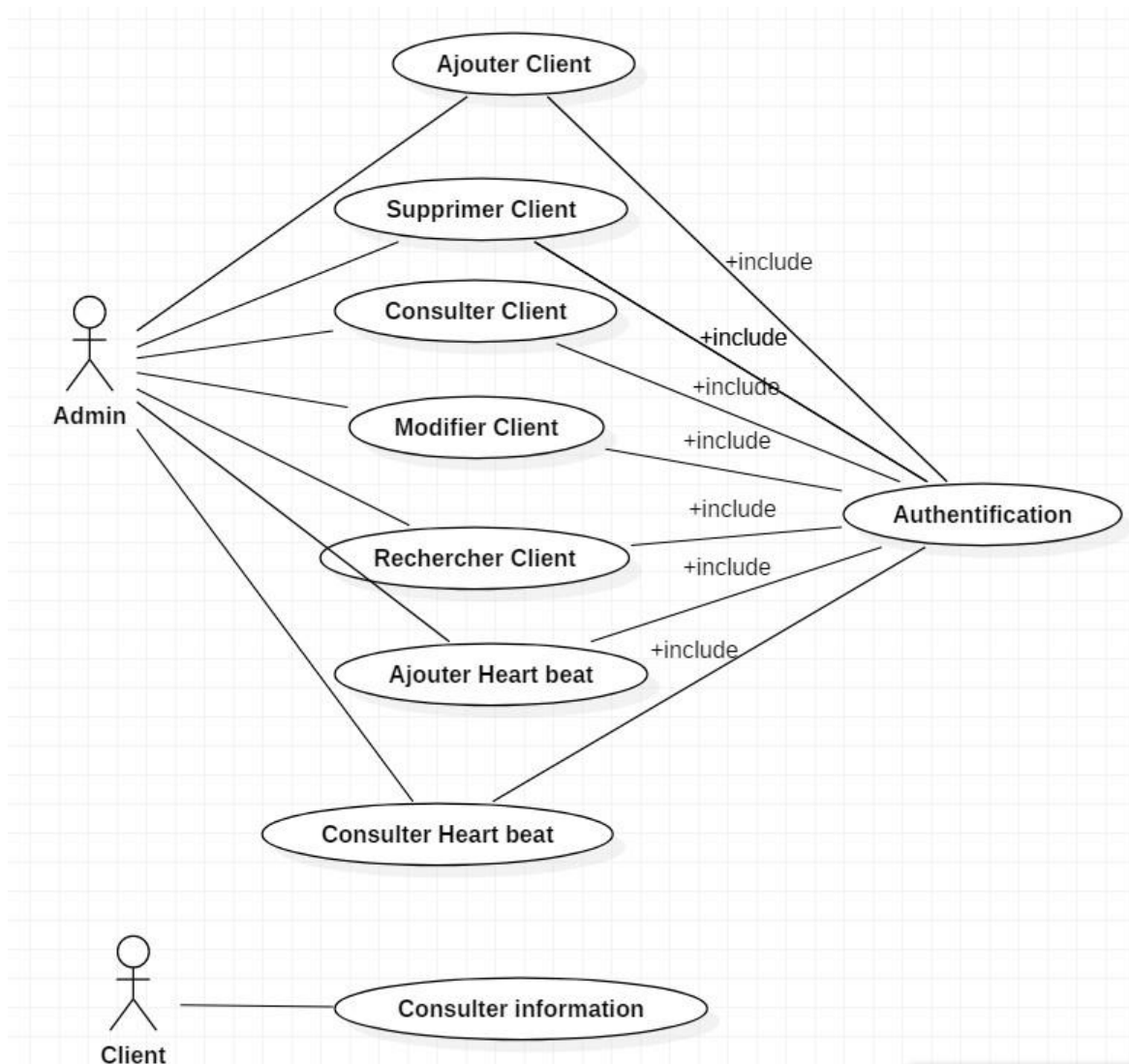


Diagramme de cas d'utilisation

2- Diagramme de séquence

Définition :

Un diagramme de séquence est un outil de modélisation qui représente l'interaction entre différents objets ou composants d'un système dans une séquence chronologique.

-Ajouter un nouveau client

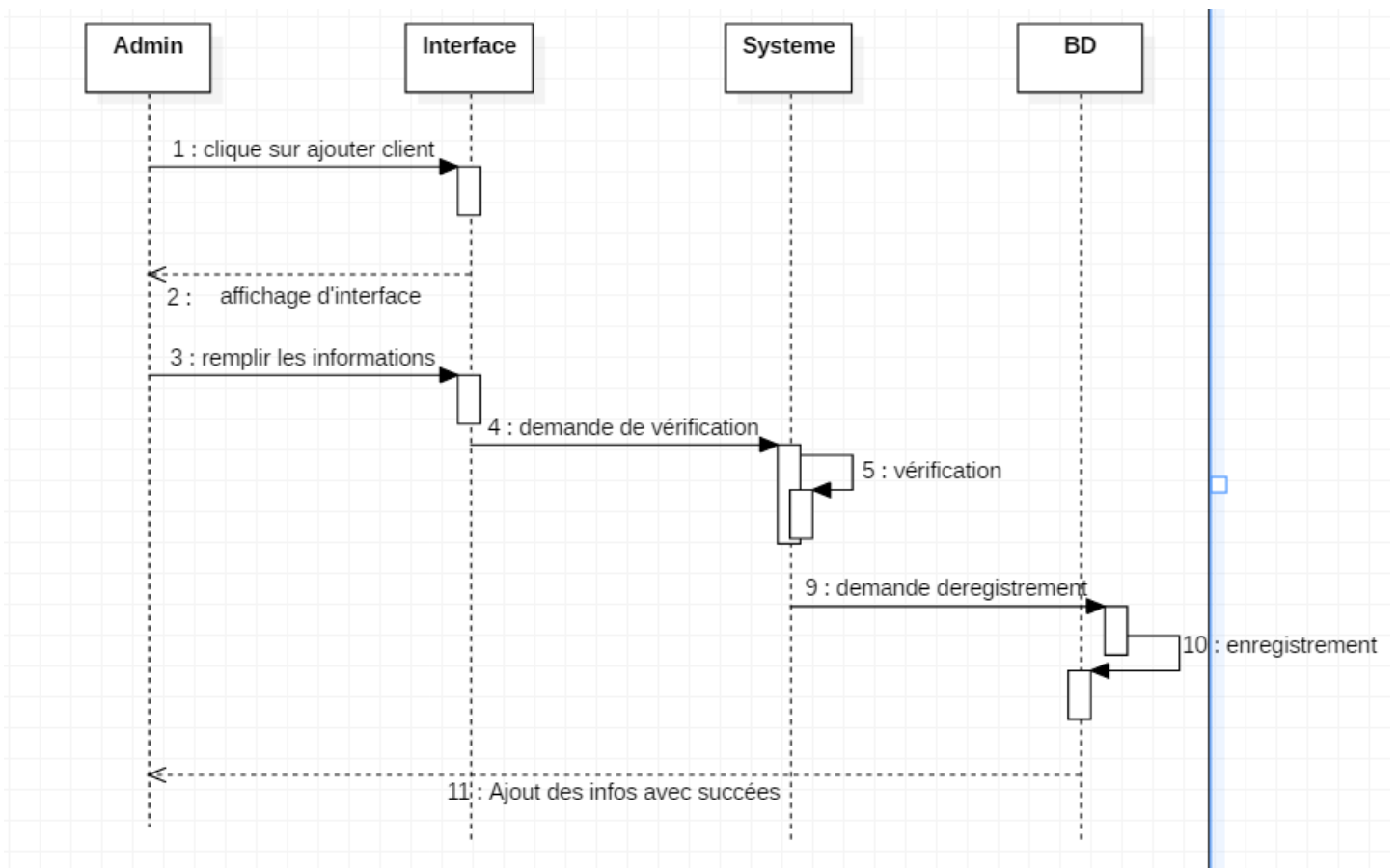


Diagramme de séquence Ajouter un nouveau client

-Authentification Client

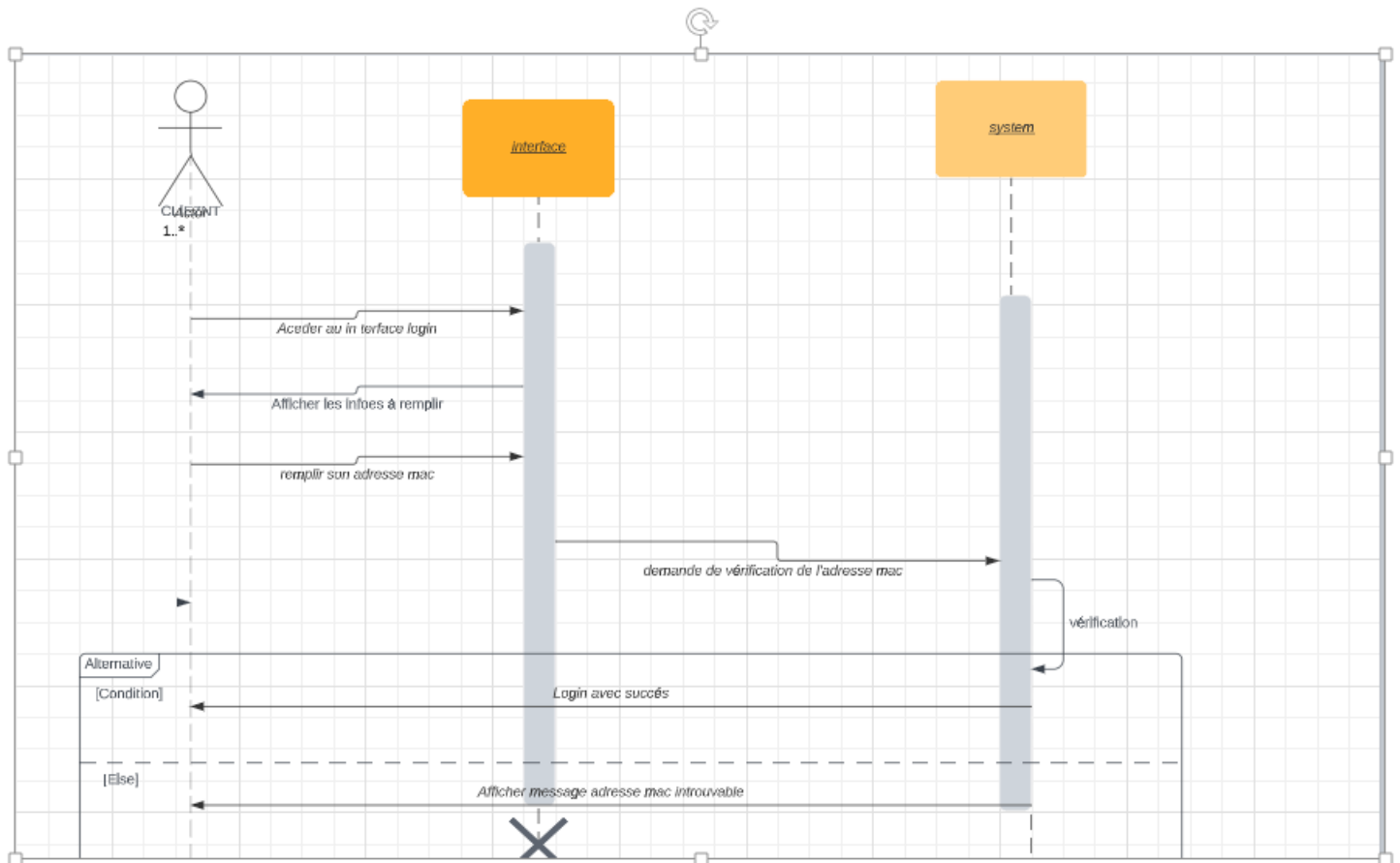


Diagramme de séquence Authentification client

-Ajouter un Heart Beat

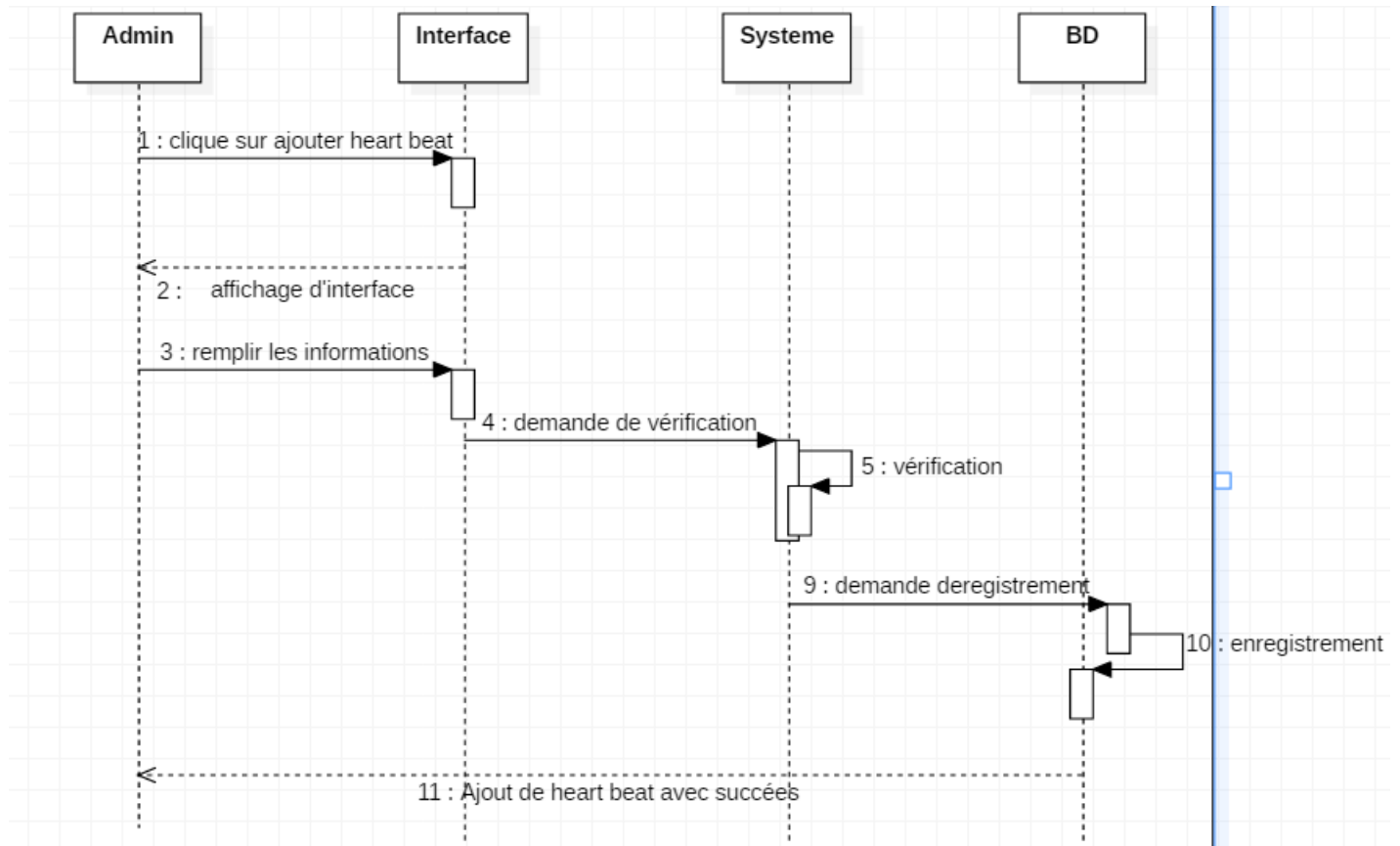
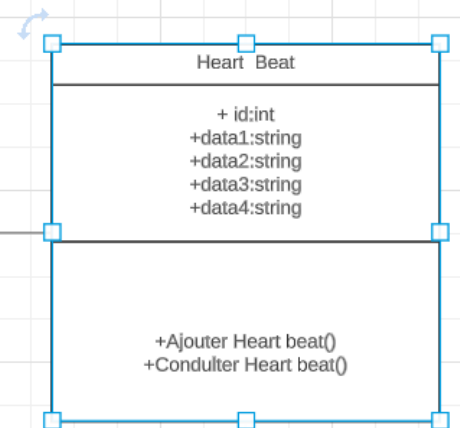
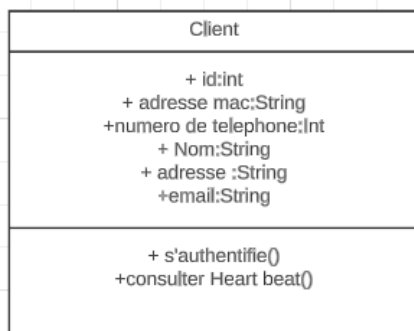


Diagramme de séquence Ajouter Heart Beat

3- Diagramme de classe

Définition :

Un diagramme de classe représente la structure statique d'un système logiciel, montrant les classes, leurs attributs et leurs méthodes, ainsi que les relations entre elles. C'est une visualisation essentielle pour comprendre la composition d'un système logiciel.



Partie 2: Contexte, Technologies et Architecture

1-Rappel du contexte générale du projet

Notre projet vise à développer une application qui exploite les données de battements de cœur provenant d'une montre connectée (smartwatch) et les affiche de manière graphique. L'objectif est de permettre aux utilisateurs de surveiller et d'analyser leurs battements de cœur de manière pratique

Cela permet aux utilisateurs de surveiller leur santé cardiaque, de détecter d'éventuelles anomalies et de prendre des mesures appropriées, le cas échéant.

L'application propose deux interfaces distinctes : une interface client et une interface administrateur. L'interface client est destinée aux utilisateurs finaux, qui peuvent consulter leurs propres données de battements de cœur et suivre leur évolution au fil du temps. L'interface administrateur permet quant à elle à un personnel autorisé de gérer les clients, d'ajouter manuellement des données de battements de cœur et de superviser l'ensemble du système.

Pour développer les deux interfaces on a utilisé le Framework angular

Le développement de l'application s'est concentré sur la création d'une interface frontend conviviale qui communique avec des API pour effectuer des opérations telles que l'ajout de clients, la suppression de clients, la consultation des données de battements de cœur....

2- Technologie et outils utilisés



Angular est un Framework open-source de développement d'applications web créé par Google, utilisant Type Script et offrant des fonctionnalités avancées pour la création d'applications interactives et réactives. De développement frontend



est une bibliothèque qui permet d'intégrer et de manipuler facilement des graphiques interactifs dans des applications Angular



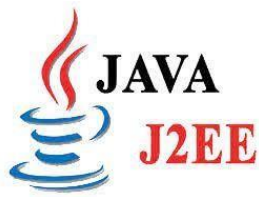
Le langage de modélisation unifié (UML) est un langage de modélisation graphique utilisé pour visualiser, spécifier, concevoir et documenter les différents aspects d'un système logiciel. Il offre une représentation visuelle standardisée des éléments clés d'un système et de leurs relations.



Postman est un outil de développement d'API qui permet aux développeurs de tester, de déboguer et de documenter les API de manière simple et efficace



Git est un système de contrôle de version qui permet de gérer les différentes versions d'un projet, tandis que GitHub est une plateforme en ligne qui utilise Git pour héberger, partager et collaborer sur des projets de développement de logiciels



Java J2EE est une plateforme de développement back end en Java pour la création d'applications d'entreprise sécurisées, évolutives et robustes.

3-Architecture

L'architecture de notre projet repose sur le framework Angular, qui suit une approche de développement front-end basée sur les composants. Nous utilisons Angular pour créer une application web conviviale qui communique avec une montre connectée pour afficher les graphiques de battements de cœur.

L'application est structurée en plusieurs composants principaux :

Le composant "ClientsComponent" est responsable de l'affichage de la liste des clients et de la gestion des opérations liées aux clients. Il utilise le service "ClientService" pour interagir avec les données des clients, notamment pour récupérer la liste des clients, ajouter de nouveaux clients et supprimer des clients existants. Ce composant affiche également une barre de recherche permettant de filtrer les clients en fonction de leur nom.

Le composant "DetailsComponent" est utilisé pour afficher les détails d'un client spécifique. Il récupère les informations du client à l'aide du service "ClientService" et affiche les données personnelles du client ainsi que les données de battements de cœur associées. Ce composant utilise également le service "HeartbeatService" pour récupérer les données de battements de cœur spécifiques à ce client.

Le composant "GrapheComponent" est responsable de l'affichage des graphiques de battements de cœur. Il utilise le package "angular-highcharts" pour créer et afficher les graphiques à partir des données de battements de cœur. Ce composant reçoit les données de battements de cœur du composant "DetailsComponent" et génère les graphiques appropriés pour les afficher à l'utilisateur.

Le composant "LoginComponent" gère l'interface de connexion de l'application. Il permet aux utilisateurs de se connecter en utilisant leur identifiant de montre connectée (adresse MAC) et redirige vers la page appropriée en fonction du type d'utilisateur (administrateur ou client). Ce composant utilise le service "AuthService" pour gérer l'authentification des utilisateurs.

Le composant "NavbarComponent" est responsable de l'affichage de la barre de navigation de l'application. Il fournit des liens vers les différentes fonctionnalités de l'application, notamment la page d'accueil, la liste des clients et la déconnexion.

En ce qui concerne l'architecture de communication, l'application suit une approche client-serveur. Les composants front-end (tels que "ClientsComponent", "DetailsComponent" et

"GrapheComponent") communiquent avec les services correspondants (tels que "ClientService" et "HeartbeatService") pour récupérer les données nécessaires à partir des API du serveur.

Les services (tels que "ClientService" et "HeartbeatService") utilisent des appels HTTP pour interagir avec le serveur backend, qui fournit les données de clients et de battements de cœur. Les services encapsulent la logique métier nécessaire pour récupérer, manipuler et persister les données.

Au niveau technologique, l'application utilise Angular comme framework front-end, ce qui permet une gestion efficace des composants, des services et de la communication avec le serveur. Les graphiques de battements de cœur sont générés à l'aide du package "angular-highcharts", qui facilite la création de graphiques interactifs.

En résumé, l'architecture de notre application repose sur le framework Angular, avec plusieurs composants interagissant avec des services pour récupérer et manipuler les données des clients et des battements de cœur. L'application suit une approche client-serveur, avec des appels HTTP pour communiquer avec le serveur backend. Cette architecture permet de créer une application conviviale et réactive, offrant aux utilisateurs la possibilité de consulter et d'analyser leurs données de battements de cœur de manière efficace.

4- Résultats

Interface admin



Bienvenue sur notre site de santé

Nous sommes dédiés à votre bien-être et à votre santé.

Grâce à notre équipe de professionnels médicaux hautement qualifiés, nous offrons une gamme complète de services de santé pour répondre à vos besoins individuels.

Que vous recherchiez des soins médicaux préventifs, des traitements spécialisés ou des conseils en matière de santé, nous sommes là pour vous accompagner.

N'hésitez pas à parcourir notre site pour en savoir plus sur nos services et à nous contacter pour prendre rendez-vous.

[Prendre rendez-vous](#)

Suivez-nous sur les réseaux sociaux :



List of clients

[+New Client](#)

Adresse MAC:

AC:41:5F:85:C8:60

Nom:

user1

Prénom:

user

Téléphone:

0645789823

Adresse:

user

Email:

user@gmail.com

[Nouveau client](#)[X Annuler](#)

List of clients

+New Client

Recherche...

user user

test test

Data1

0

Data2

0

Data3

0

Data4

0

ADD HEARTBEAT

Client Details

[+New HeartBeat](#)

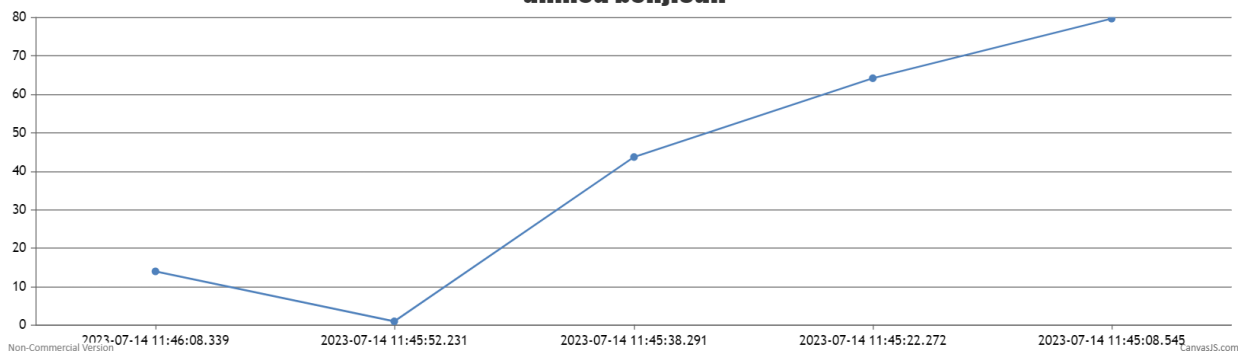
Id	33
Mac	AC:41:5F:85:C8:60
Nom	user
Prenom	user
Tel	0645789823
Adresse	user
Mail	user@gmail.com

user user



Id	35
Mac	AC:45:5F:85:C8:60
Nom	ahmed
Prenom	benjloun
Tel	0654782589
Adresse	test
Mail	ahmed@gmail.com

ahmed benjloun



Interface client

Login

AC:45:5F:85:C8:60

Login

Health is not valued
till sickness comes

Id	35
Mac	AC:45:5F:85:C8:60
Nom	ahmed
Prenom	benjloun
Tel	0654782589
Adresse	test
Mail	ahmed@gmail.com

