

# **Documentation**

## **Suivi de projet**

# **CHATY**

**Baptiste SOULIER & Djebri HARHAD**

# **Table des matières**

- 1. Table des matières
- 2. Objectifs du Projet
- 3. Cahier des charges et Fonctionnalités
- 4.Équipe de Projet
- 5. Plan de Travail
  - 5.1. Chronogramme
  - 5.2. Phases et Jalons
- 6. Technologies, Maquettes et Wireframes
- 7. Répartition des Tâches
- 8. Suivi de projet
- 9. Rapports de Progrès Mi-Projet
- 10. Décisions Charnière
- 11. Clôture du Projet
  - 11.1. Critères de Réussite
  - 11.2. Problèmes Rencontrés
  - 11.3. Leçons Apprises
- 12. Annexes

## Objectif du Projet

Le but du projet est de créer une plateforme permettant de discuter en vidéo et par texte avec son ami, sa moitié, son recruteur, son collègue, sa famille dans une salle qui vous est propre avec la plus grande simplicité sans téléchargement, directement depuis son navigateur.

## Cahier des Charges et Fonctionnalités

Nous avons quelques exigences. Notre projet doit avoir :

- Un frontend
- Un backend
- Une API
- Une infrastructure faisant fonctionner le projet

Nous avons alors décidé des fonctionnalités que nous voulons intégrer au projet. Nous les avons classés en deux colonnes :

- MVP : les fonctionnalités que Chaty doit absolument contenir.
- OPT : les fonctionnalités optionnelles que nous pourrions ajouter à Chaty une fois que nos fonctionnalités MVP seront toutes intégrées.

Fonctionnalités du projet	MVP	Optionnel
Login		
Modifier profil		
Avatar		
Chat textuel		
Option de couper la caméra et le micro		
Chat vidéo		
Créer une salle		
Système d'amitié		
Inviter des amis depuis la messagerie		
Messagerie privée		

## Equipe du Projet



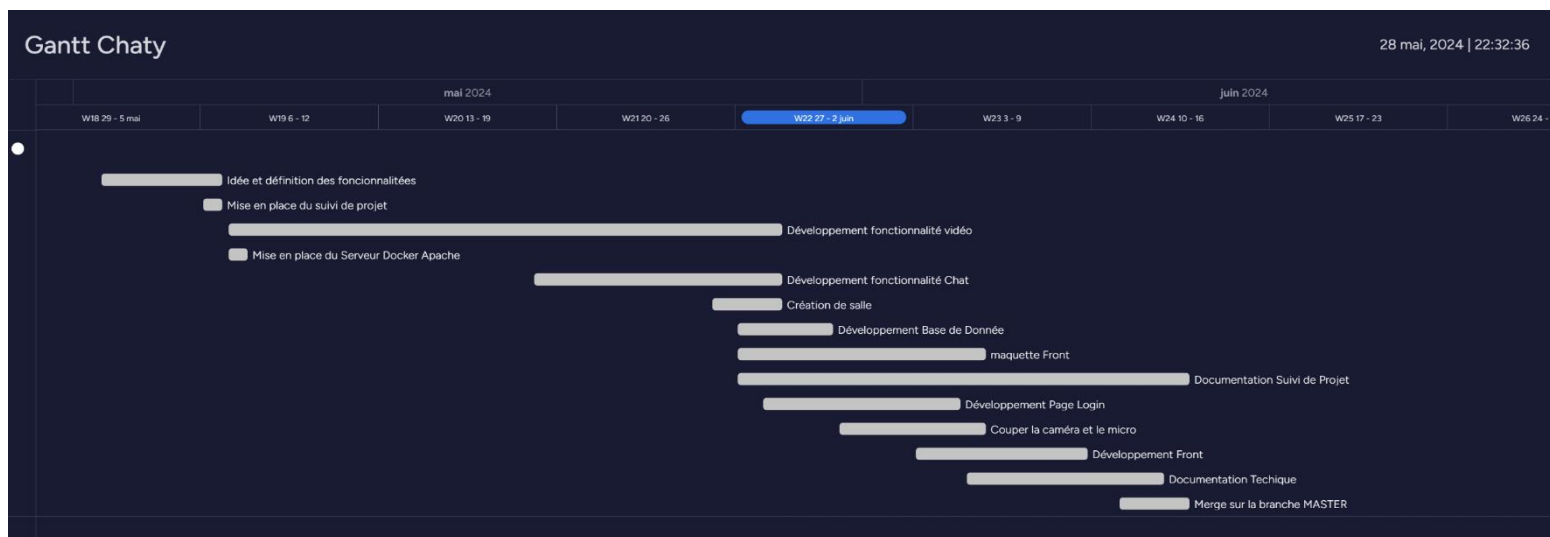
Djebri HARHAD



Baptiste SOULIER

## Plan de Travail

Diagramme de Gantt du projet Chaty de la période du 6 mai au 14 juin :

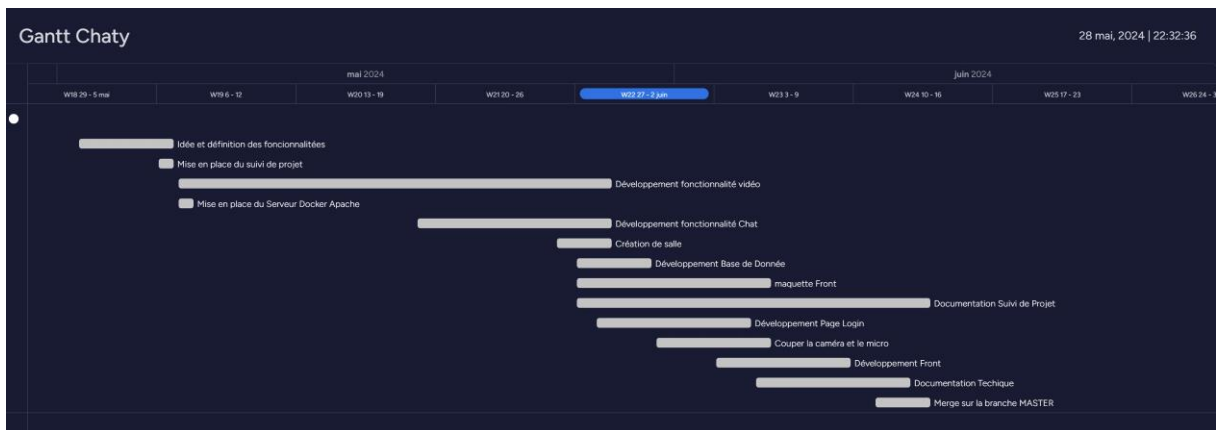


# Conception

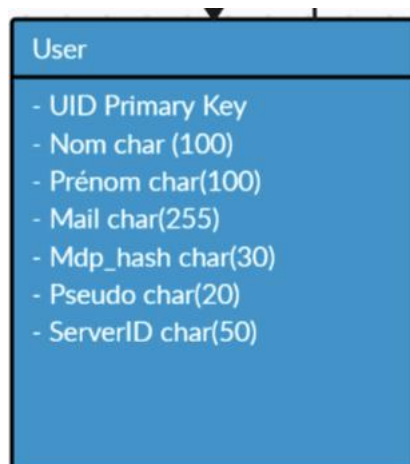
## Technologies, Maquettes et Wireframes

Pour Chaty, nous avons utilisé les technologies suivantes :

- Frontend et Backend : Javascript, HTML, CSS
- Flux Vidéo : L'API de WebRTC
- Chat textuel : Socket.io
- Base de données : PostgreSQL
- Infrastructure : Docker, Apache, server FTP



Modèle Conceptuel des Données de CHATY :



Maquette Frontend : Voir Figma en annexe

## Répartition des Tâches

Djebril HARHAD	Baptiste SOULIER
Flux Vidéo	Docker Apache
Docker pgAdmin	Docker PostgreSQL
BDD	Frontend
Docker redis	Chat Textuel
Session	Page Login
Maquette Figma	Maquette Figma
Documentation Technique	Documentation Technique
Documentation Gestion de Projet	Documentation Gestion de Projet

## Suivi de Projet

Pour suivre l'évolution de notre projet, nous avons utilisé Monday. Cette plateforme a été utile afin d'avoir une vue sur notre planning et maîtriser le temps de mise en place des fonctionnalités nécessaires.

Nous avons travaillé principalement en physique dans nos bureaux à Paris Ynov Campus ; nous nous sommes fixés deux réunions hebdomadaires pour échanger et analyser régulièrement sur notre réalisation afin d'améliorer Chaty.

Evidemment le télétravail était aussi présent donc nous avons utilisé Microsoft Teams et Discord pour communiquer quotidiennement.



## Rapport de Progrès

Rapport de progrès mi-projet : 29/05/24 :

Nous sommes arrivés à mi-projet. Il est temps de faire un point sur les fonctionnalités que nous avons terminées, celles que nous devons absolument finir ou commencer et voir si nous sommes dans les temps ou en retard par rapport à notre retroplanning afin revoir notre ordre de priorité.

Nom de la tâche	Priorité	Statut
▼ À faire		
✓ Système d'amitié	Optionnel	
✓ Inviter depuis la messagerie	Optionnel	
✓ Messagerie Privée	Optionnel	
✓ Frontend	Moyenne	Dans les délais
✓ Documentation Technique	Moyenne	Dans les délais
✓ Modifier le Profil	Moyenne	Dans les délais
✓ Développement Login User	Élevée	Dans les délais
✓ Server FTP	Élevée	Dans les délais
✓ Couper le Micro et la Caméra	Élevée	En retard
Ajouter une tâche...		
▼ En cours		
✓ Documentation suivi de projet	Moyenne	Dans les délais
✓ Base de donnée en postgresQL depuis Docker	Moyenne	Dans les délais
✓ Maquette Frontend	Faible	Dans les délais
Ajouter une tâche...		
▼ Terminé		
✓ Développement du Vidéo	Élevée	Terminé
✓ Développement Chat	Élevée	Terminé
✓ Server Docker Apache	Élevée	Terminé
✓ Création de salle	Élevée	Terminé
✓ Modèle Conceptuel des Données	Faible	Terminé

## Décisions Charnière

La section "Décisions Charnière" est cruciale pour garder une trace des décisions majeures prises au cours du projet. La principale décision était le choix des technologies. Plus haut dans cette documentation de suivi de projet nous avons énumérer les technologies que nous avons utilisé. Dans cette rubrique nous expliquons ce choix.

### Choix des Technologie :

- Node.js : Une plateforme JavaScript côté serveur qui permet d'utiliser JavaScript pour écrire des applications côté serveur. Parfait pour les applications en temps réel grâce à sa nature non-bloquante.
- PostgreSQL : Une base de données relationnelle open-source puissante, avec une forte conformité aux standards SQL. Idéal pour des données relationnelles complexes.
- Socket.IO : Facilite la communication en temps réel entre le serveur et le client. Parfait pour les fonctionnalités de chat en temps réel.
- WebRTC : Une API qui permet la communication audio et vidéo en temps réel directement entre les navigateurs. Essentiel pour les appels vidéo et audio en temps réel.
- Docker : Docker offre une solution légère, portable et efficace pour le développement et le déploiement d'applications. Il facilite la gestion des dépendances et assure une cohérence des environnements de développement, de test et de production, tout en optimisant l'utilisation des ressources matérielles.







## Clôture du projet

Chaty est désormais terminé, voici notre retour sur nos critères de réussite, nos problèmes rencontrés, ce que nous avons appris et la suite du projet.

### Critères de réussite :

Nous nous sommes posés des critères de réussite en fonction du résultat obtenu à la fin de la conception de Chaty :

- Avoir une application fonctionnelle 
- Intégrer la totalité de nos fonctionnalités prioritaires 
- Utilisation de Docker pour l'hébergement du serveur, de la base de données et de Pgadmin. 
- Avoir une interface et une bonne accessibilité 

Nous sommes fiers de notre réalisation et estimons avoir remplis tous les critères imposés.

### Problèmes rencontrés :

Forcément, durant le développement de Chaty nous avons rencontré un gros problème : la transmission de la data vidéo. Après recherche du problème nous avons repéré que la data ne s'envoyait pas car la direction était mauvaise. Nous avons pu régler le problème sans prendre de retard sur notre retroplanning.

### Leçon Apprises :

La diversité et la nouveauté de certaines technologies que nous avons décidé d'utiliser nous a amené à apprendre à les utiliser, à les comprendre.

- WebRTC
- Socket.IO
- Docker

### Suite du projet :

Nous sommes fiers de notre réalisation et continuerons à améliorer Chaty dans un futur proche. Tout d'abord en incluant nos fonctionnalités optionnelles puis en améliorant la capacité des salles afin de pouvoir communiquer à plusieurs dans une salle. Ensuite nous améliorerons l'expérience et l'interface utilisateur et pour finir nous améliorerons la sécurité de la plateforme Chaty. Un beau projet qui verra le grand jour dans quelques années.

## **Annexes**

Vous pourrez trouver en annexes de ce dossier :

- Diagramme de Gant
- Le Modèle Conceptuel des données de Chaty
- Le PDF de la Maquette Front – end
- Le lien du GitHub de Chaty : [BaptSoulter/chat-app \(github.com\)](https://github.com/BaptSoulter/chat-app)