|  |
| --- |
| Projet WebChat |

Table des matières

[1 Analyse préliminaire 3](#_Toc499021832)

[1.1 Introduction 3](#_Toc499021833)

[1.2 Objectifs 3](#_Toc499021834)

[1.3 Planification initiale 4](#_Toc499021835)

[2 Analyse / Conception 4](#_Toc499021836)

[2.1 Concept 4](#_Toc499021837)

[2.2 Stratégie de test 4](#_Toc499021838)

[2.3 Risques techniques 4](#_Toc499021839)

[2.4 Planification 4](#_Toc499021840)

[2.5 Dossier de conception 5](#_Toc499021841)

[3 Réalisation 5](#_Toc499021842)

[3.1 Dossier de réalisation 5](#_Toc499021843)

[3.2 Description des tests effectués 6](#_Toc499021844)

[3.3 Erreurs restantes 6](#_Toc499021845)

[3.4 Liste des documents fournis 6](#_Toc499021846)

[4 Conclusions 6](#_Toc499021847)

[5 Annexes 7](#_Toc499021848)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 7](#_Toc499021849)

[5.2 Sources – Bibliographie 7](#_Toc499021850)

[5.3 Journal de travail 7](#_Toc499021851)

[5.4 Manuel d'Installation 7](#_Toc499021852)

[5.5 Manuel d'Utilisation 7](#_Toc499021853)

[5.6 Archives du projet 7](#_Toc499021854)

# Analyse préliminaire

## Introduction

*Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Objectifs

*Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Planification initiale

*Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.*

*Ces éléments peuvent être repris des spécifications de départ.*

# Analyse / Conception

## Concept

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

* *Multimédia: carte de site, maquettes papier, story board préliminaire, …*
* *Bases de données: interfaces graphiques, modèle conceptuel.*
* *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle…*
* *…*

## Stratégie de test

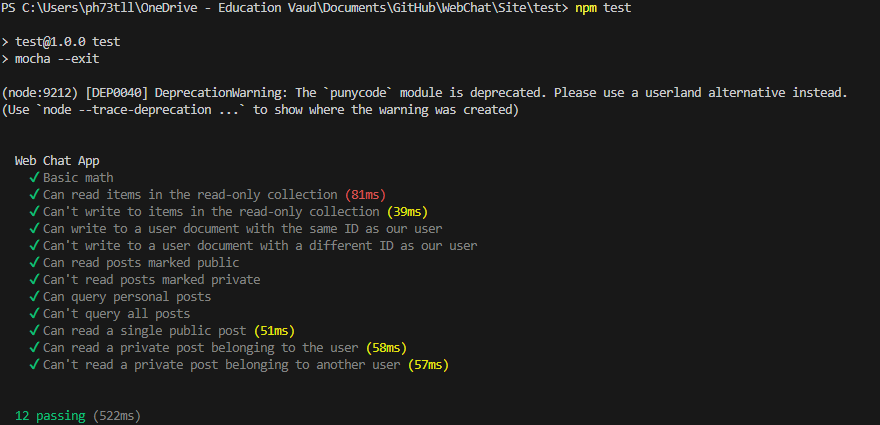
Bien que Jest soit un framework populaire pour les tests JavaScript, j'ai préféré utiliser la Firebase Emulator Suite pour plusieurs raisons spécifiques à Firebase.

1. Intégration étroite avec Firebase : La suite d'émulateurs est spécifiquement conçue pour fonctionner avec les services Firebase, offrant ainsi une expérience de test cohérente et précise pour ces fonctionnalités.
2. Environnement contrôlé : Les tests peuvent être exécutés localement dans un environnement isolé, garantissant que les données réelles de l'application ne sont pas affectées. Cela réduit les risques de corruptions de données ou d'effets secondaires indésirables pendant les tests.
3. Facilité d'utilisation : La Firebase Emulator Suite est conviviale et facile à configurer, ce qui permet aux développeurs de se concentrer sur l'écriture de tests efficaces plutôt que sur la mise en place de l'infrastructure de test.

L'utilisation de la Firebase Emulator Suite garantit que les fonctionnalités de l'application fonctionnent correctement avant d'être déployées en production. Cela permet de détecter et de corriger les erreurs potentielles plus tôt dans le processus de développement, ce qui contribue à améliorer la qualité globale de l'application et à réduire les risques d'erreurs après le déploiement.

En résumé, la Firebase Emulator Suite offre une solution pratique et intégrée pour tester les fonctionnalités Firebase localement, ce qui permet de garantir la qualité et la fiabilité de l'application avant son déploiement en production.

Pour commencer, j'ai téléchargé Java 17, car la Firebase Emulator Suite nécessite Java 11 ou ultérieur pour fonctionner correctement. Ensuite, j'ai installé Mocha en utilisant la commande npm install mocha --save-dev et le package @firebase/testing avec la commande npm install @firebase/testing --save-dev.

Voici la liste des tests :

## Risques techniques

* *risques techniques (complexité, manque de compétences, …).*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, …).*

## Planification

*Révision de la planification initiale du projet :*

* *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
* *partage des tâches en cas de travail à plusieurs.*

*Il s’agit en principe de la planification* ***définitive du projet****. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.*

## Dossier de conception

### Firebase

C'est une plateforme de développement d’applications web et mobiles soutenue par Google. On parle de Backend As A Service (BaaS). Firebase gère sa propre infrastructure avec un bel ensemble d’outils pour simplifier le travail du développeur en lui fournissant des kits de développement et un tableau de bord en ligne.

### Les prix

|  |  |
| --- | --- |
| No-cost | Pay as you go |
| Spark Plan | Blaze Plan |
| Quotas d'utilisation pour Database, Firestore, Storage, Phone Auth, Hosting et Test Lab | Inclut une utilisation sans frais, calculée chaque jour. Vous ne payez ensuite que ce que votre projet utilise. |

Pour plus de détails [firebase/pricing](https://firebase.google.com/pricing).

### Concurrents

1. [AWS Amplify](https://www.theknowledgeacademy.com/blog/what-is-aws-amplify/) :

* Base de données : AWS Amplify offre une gamme de services cloud, y compris Amazon DynamoDB, une base de données NoSQL, pour stocker les données. Bien qu'Amplify ne propose pas de base de données en temps réel comme Firebase, DynamoDB est flexible et évolutif.
* Authentification : AWS Amplify offre des services d'authentification via Amazon Cognito, qui prend également en charge diverses méthodes d'authentification et de gestion des utilisateurs.
* Intégration : Amplify s'intègre étroitement avec d'autres services AWS, offrant une suite complète de solutions cloud pour les entreprises déjà investies dans l'écosystème AWS.

1. [Microsoft Azure Mobile Apps](https://learn.microsoft.com/en-us/azure/developer/mobile-apps/azure-mobile-apps/overview) :

* Base de données : Azure Mobile Apps utilise Microsoft SQL Database, une base de données SQL relationnelle, pour stocker les données. SQL Database offre une structure relationnelle plus traditionnelle. Et utilise aussi une base de données NoSQL (Cosmos DB).
* Authentification : Azure Mobile Apps propose une authentification via Azure Active Directory, offrant une gestion centralisée des identités et des accès.
* Intégration : Azure Mobile Apps s'intègre naturellement avec d'autres services Azure, ce qui peut être préférable pour les organisations déjà engagées dans l'écosystème Microsoft.

1. [Backendless](https://backendless.com/) :

* Base de données : Backendless propose une base de données cloud NoSQL flexible et hautement scalable, adaptée aux différentes structures de données des applications. Elle offre une gestion simplifiée et une évolutivité facile.
* Authentification : Backendless fournit des fonctionnalités d'authentification robustes, incluant diverses méthodes d'authentification (e-mail, réseaux sociaux) et la gestion des utilisateurs et des rôles pour sécuriser les applications.
* Intégration : Backendless s'intègre facilement avec des services tiers et des API, offrant des SDK pour différentes plateformes de développement. Cette flexibilité permet aux développeurs d'élargir les fonctionnalités de leurs applications.

### Comparaison des technologies

#### Voici les principaux types de bases de données NoSQL :

* Bases de données de documents :
  + Stockent les données sous forme de documents, généralement au format JSON, XML ou BSON.
  + Chaque document peut contenir des champs et des valeurs de données de manière flexible.
  + Exemples : MongoDB, firestore(firebase).
* Bases de données de paires clé-valeur :
  + Stockent les données sous forme de paires clé-valeur simples.
  + Les valeurs peuvent être de simples chaînes de caractères, des structures de données plus complexes ou même des objets binaires.
  + Exemples : Redis, Amazon DynamoDB.
* Bases de données de colonnes :
  + Stockent les données de manière orientée colonnes plutôt que lignes.
  + Les colonnes sont regroupées en familles de colonnes et peuvent être ajoutées dynamiquement.
  + Exemples : Apache Cassandra, HBase.
* Bases de données de graphes :
  + Conçues pour stocker et traiter des données hautement interconnectées sous forme de graphes.
  + Utilisent des structures de données graphiques pour représenter les entités et les relations entre elles.
  + Exemples : Neo4j, Amazon Neptune.

#### Popularité + Tendance

Firebase est largement utilisé dans le développement d'applications mobiles et web, en raison de sa simplicité d'utilisation, de sa robustesse et de son intégration étroite avec d'autres services Google. Il et utilisé par des journaux comme "The New York Times" et "The Economist", des applications de messagerie comme "Whatsapp" et "Instagram" et bien d'autres applications (Twitter,SoundCloud,Airbnb)

##### Quelques chiffres pris depuis [StackShare](https://stackshare.io/firebase" \l "stacks) et [builtWith](https://trends.builtwith.com/websitelist/Firebase):

* 3'078 entreprises utiliseraient Firebase dans leurs piles technologiques.
* 35'502 développeurs ont déclaré qu'ils utilisaient Firebase.
* 319'563 site webs utiliseraient actuellement firebase.

*Fournir tous les document de conception:*

* *le choix du matériel HW*
* *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
* *le choix des outils logiciels pour la réalisation et l'utilisation*
* *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, …*
* *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
* *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

# Réalisation

## Dossier de réalisation

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Evitez d’inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez que cette partie…*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## 

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*