|  |  |
| --- | --- |
| C:\Users\FatmaZ\Desktop\stage\rapport\logo_univrouen.jpg | C:\Users\FatmaZ\Desktop\stage\rapport\téléchargement.png |

**RAPPORT DE STAGE**

**Développeur Informatique**

Effectué par: ZEKRI Fatma

01 Avril 2019 – 24 Mai 2019

|  |  |
| --- | --- |
| **Tuteur de stage :** BESSONNET Julien | **Enseignant référant**: HERAUVILLE Stéphane |
| **Entreprise d’accueil :** YuSit | **Etablissement :** Université de Rouen  **Diplôme :** Licence Informatique |

Table des matières

[**1.** **Remerciements :** 4](#_Toc10511376)

[**2.** **Résumé :** 5](#_Toc10511377)

[**3.** **Introduction :** 5](#_Toc10511378)

[**3.1.** **Présentation de l’entreprise :** 5](#_Toc10511379)

[**3.2.** **Présentation de l’équipe :** 5](#_Toc10511380)

[**3.3.** **Les services proposés par l’entreprise :** 6](#_Toc10511381)

[**3.4.** **Présentation du projet :** 6](#_Toc10511382)

[**3.5.** **Principe du jeu « mots fléchés » :** 7](#_Toc10511383)

[**4.** **Analyse du projet :** 7](#_Toc10511387)

[**4.1.** **Introduction :** 7](#_Toc10511388)

[**4.2.** **Spécifications des besoins :** 7](#_Toc10511389)

[**4.3.** **Diagramme de Cas d’Utilisation :** 8](#_Toc10511390)

[**4.4.** **Les outils utilisés :** 11](#_Toc10511391)

[**4.4.1.** **Visual Studio Code :** 11](#_Toc10511392)

[**4.4.2.** **Angular :** 11](#_Toc10511393)

[**4.4.3.** **FireBase :** 14](#_Toc10511394)

[**5.** **Développement de l’application:** 15](#_Toc10511395)

[**5.1.** **Les Composants :** 15](#_Toc10511396)

[**5.2.** **Les Modèles :** 16](#_Toc10511397)

[**5.3.** **Les Services :** 17](#_Toc10511398)

[**6.** **Réalisation :** 17](#_Toc10511399)

[**6.1.** **Partie Back-end :** 17](#_Toc10511400)

[**6.2.** **Partie Front-end:** 18](#_Toc10511401)

[**7.** **Conclusion :** 20](#_Toc10511402)

[En effet le stage consistait à réaliser une application de génération de mots fléchés avec en Angular et sauvegarder les données sur une base de données Firebase. Il m’a permis d’améliorer mes connaissances en création d’application web, et notamment en ce qui concerne le respect de l’architecture logicielle. J’ai également appris l’utilisation du framework Angular avec lequel j’ai développé mon application. 20](#_Toc10511403)

[A l’heure actuelle, l’application est prête à être utilisée. On peut donc affirmer que le but qui m’avait été fixé a été atteint. 20](#_Toc10511404)

[En conclusion, ce stage est complémentaire à mon année d’étude au sein de la formation Licence Informatique. Il m’a apporté une nouvelle expérience, des nouvelles connaissances professionnelles et humaines et m’a permis de comprendre les enjeux d’un projet informatique. 20](#_Toc10511405)

[**8.** **Annexes :** 21](#_Toc10511406)

[**9.** **Glossaire :** 23](#_Toc10511407)

[**10.** **Bibliographie :** 24](#_Toc10511408)

[ Documentation Angular: https://angular.io/docs 24](#_Toc10511409)

[ Tutoriel Angular: https://openclassrooms.com/fr/courses/4668271-developpez-des-applications-web-avec-angular 24](#_Toc10511410)

[ Cours IHM :L3 Informatique Université de Rouen 24](#_Toc10511411)

[ https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil\_principal 24](#_Toc10511412)

1. **Remerciements :**

Je tiens à remercier tous mes enseignants du Département Informatique à l’Université de Rouen, qui m’ont fourni les outils nécessaires à la réussite de mes études universitaires.

Et je remercie mon maitre de stage Monsieur JULIEN BESONNET, de m’avoir ouvert les portes de leur entreprise et de m’avoir pleinement accordé sa confiance et d’avoir partagé avec moi cette expérience enrichissante au sein de la société YUSIT durant 8 semaines.

1. **Résumé :**
2. **Introduction :**
   1. **Présentation de l’entreprise :**

Yusit est un bureau d’études innovant, spécialiste du patrimoine.

Forte des 20 années d’expériences de ses fondateurs dans le domaine patrimonial et culturel tant en France qu’à l’étranger, l’entreprise YuSit met son expertise au service du patrimoine. La maitrise des problématiques inhérentes aux politiques culturelles, et les compétences pluridisciplinaires de YuSit en font l’un des leaders de la création et du développement d’outils innovants de médiation et de valorisation du patrimoine culturel.

* 1. **Présentation de l’équipe :**

YuSit est une société constituée de 6 personnes regroupant associés et salariés :

* Mr. Julien BESSONNET : Responsable des projets numériques
* Mr. Stéphane KERVARREC : Développeur Informatique
* Mme Claire DELALANDE : Gérante, Archéologue et spécialiste en gestion du patrimoine
* Mme Jasmine BOUDEAU : Archéologue et spécialiste en gestion du patrimoine
* Mr. Frédéric LEBIS : Archéologue et spécialiste en gestion du patrimoine
  1. **Les services proposés par l’entreprise :**

L’équipe de YuSit développe et coordonne depuis vingt ans des projets culturels.

Elle propose notamment :

* La valorisation du patrimoine :
  + Modélisation de sites archéologiques et des œuvres.
  + Création de parcours de visites
  + Solution de mise en accessibilité
  + Création et animation de site web
  + Etude historique/ archéologique
* Médiation Culturelle :
  + Création de contenus (textes, photos, vidéo, 2D, 3D)
  + Support de visite (print ou numérique) : carnets, applications d’aide à la visite
  + Création d’interaction avec le public
  + Communication culturelle / digitale
* Ingénierie culturelle / patrimoniale :
  + Etude des publics / Evaluation de politique culturelle
  + Inventaire de faisabilité
  + Marketing culturel
  + Publications
  + Création de base de données
  1. **Présentation du projet :**

Le projet consiste à développer une application web côté serveur qui permet de générer une grille de mots fléchés.

Cette application a pour objectif de permettre à un utilisateur de créer un compte ou de se connecter en tant qu’administrateur.

Un administrateur peut gérer les mots dans la base de données en les ajoutant, supprimant ou les modifiant. Il peut aussi jouer et afficher la solution de la grille.

* 1. **Principe du jeu « mots fléchés » :**

Les mots fléchés sont un jeu de lettres, cousin des mots croisés, dont le but consiste à placer des lettres de mots dans des cases d’une grille.

Dans la grille se trouve des cages qui contiennent des définitions , des métaphores ou des synonymes munis d’une flèche chacun pour indiquer l’orientation des mots qu’on va y mettre, en verticale ou en horizontale.

La difficulté du jeu réside dans le fait que les mots qu’on doit insérer dans une ligne de cases doit coïncidé aux nombre de ces case et que les lettres de ce mots doivent aussi coïncidé avec la terminaison, le début, ou le milieu des lettres d’un autre mot formés à partir d’une autre ligne de case généralement verticale et horizontale.

1. **Analyse du projet :**
   1. **Introduction :**
   2. **Spécifications des besoins :**

L’application contient deux parties :

* Back-end :

Cette partie a pour but de permettre :

* + L’authentification via une adresse email et un mot de passe
  + Sauvegarder les mots et leurs définitions dans la base de données.
* Front-end :

Le Front-end présente la plus grande partie de l’application et permet :

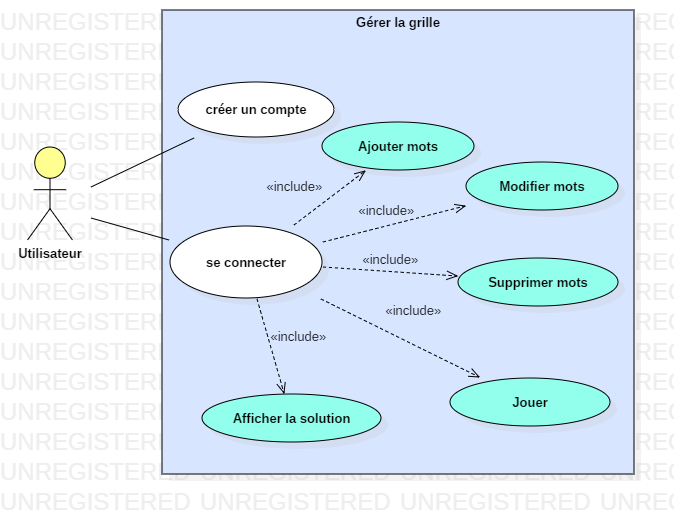
* + La création d’un compte
  + Authentification
  + Afficher la liste des mots
  + Ajouter des mots
  + Supprimer des mots
  + Modifier des mots
  + Générer une grille de mots croisés vide
  + Jouer
  + Afficher la solution
  1. **Diagramme de Cas d’Utilisation :**

En langage UML, les diagrammes de cas d’utilisation modélisent le comportement d’un système et permet de capturer les exigences du système.

Les diagrammes de cas d’utilisation décrivent les fonctions générales et la portée d’un système.

Un cas d’utilisation représente une unité discrète d’interaction entre un utilisateur (humain ou machine) et un système. Dans un diagramme de cas d’utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d’utilisation(use cases).

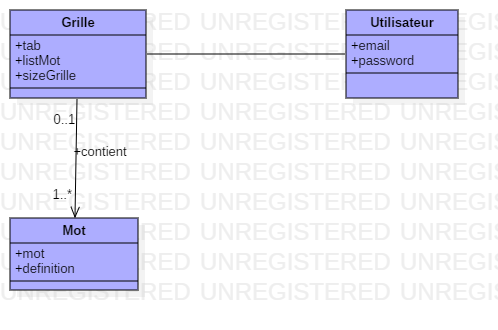
* Un utilisateur peut créer un compte.
* Un utilisateur enregistré peut :
  + Se connecter
  + Consulter la liste des mots dans la base de données
  + Ajouter des mots
  + Modifier des mots
  + Supprimer des mots
  + Jouer
  + Afficher la solution



**Diagramme des Classes :**

Ensuite, j’ai élaboré le diagramme des classes correspondant à l’application. Il représente la structure d’une application orientée objet en montrant les classes et les relations qui s’établissent entre elles.

Le diagramme de classes représente la structure d’une application orientée objet en montrant les classes et les relations qui s’établissent entre elle. Il est considéré comme le plus important de la modélisation orientée objet.

****

Description du diagramme des classes :

Les trois classes principales de cette application sont :

* Administrateur : représente le principal acteur de l’application, cette classe va être gérer par la base de données (Firebase propose un service de gestion des utilisateurs automatique). Un administrateur peut se connecter via une adresse email et un mot de passe ;
* Mot : représente le mot qui va être rentré par l’utilisateur et enregistré dans la base de données. Elle a comme attributs : mot et définition.
* Grille : représente la grille des mots fléchés qui va être générée par l’utilisateur à partir de la liste des mots enregistrés dans la base. Elle a comme attributs :
* Tab : matrice : Tableau de tableau
* listMots : liste des mots placés dans la grille
* sizeGrille : taille de la grille
  1. **Les outils utilisés :**
     1. ******Visual Studio Code :**

Visual Studio Code est un éditeur de code open-source développé parMicrosoft supportant un très grand nombre de langage grâce à des extensions.

Il supporte l’auto-compilation, la coloration syntaxique, le débugage, et les commandes des Git.

* + 1. **Angular :**

****Angular est un framework Javascript qui permet de réaliser des applications web dynamique de type « Single Page Application ». Il est gratuit, open source et créé par Google.

Les avantages d’Angular sont :

* Créer par Google : peu de chances qu’il disparaisse.
* « Type Script » : ce langage permet un développement beaucoup plus stable, rapide et facile.
* Permet le développement des applications responsives c’est-à-dire qui s’adaptent au format du support.

Un framework complet.

Utiliser Angular possède un certain nombre d’avantages, mais également des inconvénients tel que :

* Les problèmes de migration qui peuvent apparaitre lors du passage de l’ancienne version aux plus récentes.
* Les vieux navigateur ne sont pas supportés.
* Pas de version définitive/ manque de stabilité

C’est lourd surtout pour le développement des grandes applications.

* + - 1. **Architecture Angular :**
* **Architecture MVC  (Model-View-Controller):**

Au sein d’une application développée selon une architecture MVC, le code est organisé en trois parties distinctes qui collaborent étroitement. Elles concernent respectivement la gestion des données (modèle) le rendu graphique (vue) et la communication entre l’application et l’utilisateur (contrôleurs).

**Le modèle :** gère la logique applicative du logiciel, c’est-à- dire qu’il encapsule les données nécessaires à l’application, le moyen d’y accéder et la manière de les modifier.

**La vue :** gère le rendu graphique de l’application, c’est –à-dire qu’elle spécifie la manière dont il faut afficher les données que contient le modèle.

**Le contrôleur :** définit le comportement de l’application. Il reçoit les actions de l’utilisateur qui lui sont transmises par la vue, récupère éventuellement des informations à partir de la vue et traduit ces actions ainsi paramétrées par des modifications adaptées du modèle.

* **Architecture MVVM (Model-View- ViewModel):**

Le pattern MVVM est un dérivé du modèle MVC. Il a spécialement été conçu pour améliorer la séparation entre les données et la vue qui les affichent.

La Vue n’a donc plus aucun lien avec le Model. Ainsi le ViewModel s’occupe entièrement du cycle de modification de ce dernier. Il réalise à la fois la réception et l’envoi des données à la Vue, c’est le Data Binding ». Pour rappel, le binding est un mécanisme qui permet de faire des liaisons entre des données de manière dynamiques. Ce qui veut dire que si A et B sont liés, le fait de modifier A va être répercuté sur B et inversement.

Ce dernier mécanisme est la clef du pattern MVVM. Il permet de découper notre application en trois parties :

**Modèle :** contient les données.

**Vue :** La vue correspond à ce qui est affiché. Elle contient les différents composants graphiques.

**Vue-Modèle :** ce composant fait le lien entre le modèle et la vue. Il s’occupe de gérer les liaisons de données et les éventuelles conversions. C’est ici qu’intervient le binding.

* **Angular : modèle MVC ou modèle MVVM ?**

Angular est basé sur une architecture orientée composant. Il ne suit pas le modèle MVC cependant il met en œuvre un grand nombre de concepts du modèle MVVM, donc il suit le principe de MVVM.

* + - 1. **Les composants d’une application angular :**

Le diagramme suivant montre les principaux composants d’une application Angular.

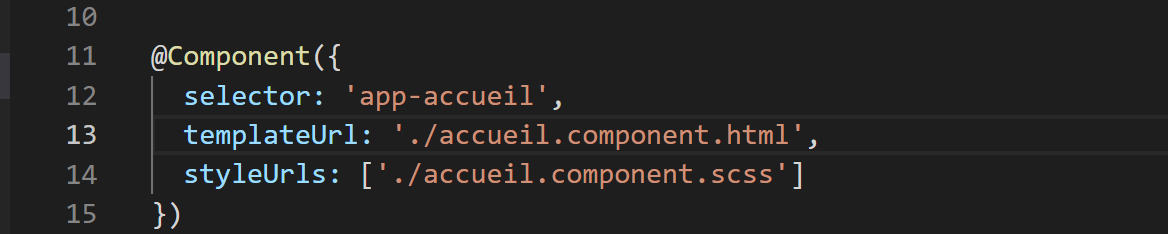


1. Component : un élément graphique composé de :

* Nomcomponent.css (feuille de style)
* Nomcomponent.html (Vue / Template)
* Nomcomponent.ts (Vue-Modèle)
* Nomcomponent.spec.ts (test)

1. Service : est une classe (.ts) qui peut être réutiliser partout dans une application. Les services peuvent servir à différentes actions comme communiquer avec une base de données, implémenter un filtre ou encore stocker des fonctions.
2. Data Binding : permet la communication entre le code TypeScript et le template HTML. Il existe trois types de Binding :

* Event Binding : permet à l’application de répondre aux actions de l’utilisateur sur une page HTML (cliquer sur un bouton).
* Property Binding : permet d’afficher des données venant du code TypeScript sur le navigateur.
* Two-way Binding: utilise Event Binding et Property Biding au même temps.

1. Metadata : Un décorateur de classe est utilisé pour attacher des métadonnées à une classe.

Les directives : est un élément HTML personnalisé utilisé pour étendre la puissance du HTML ;

Les directives structurelles : ngIf, ngFor.

Les directives par attribut : ngStyle, ngClass.

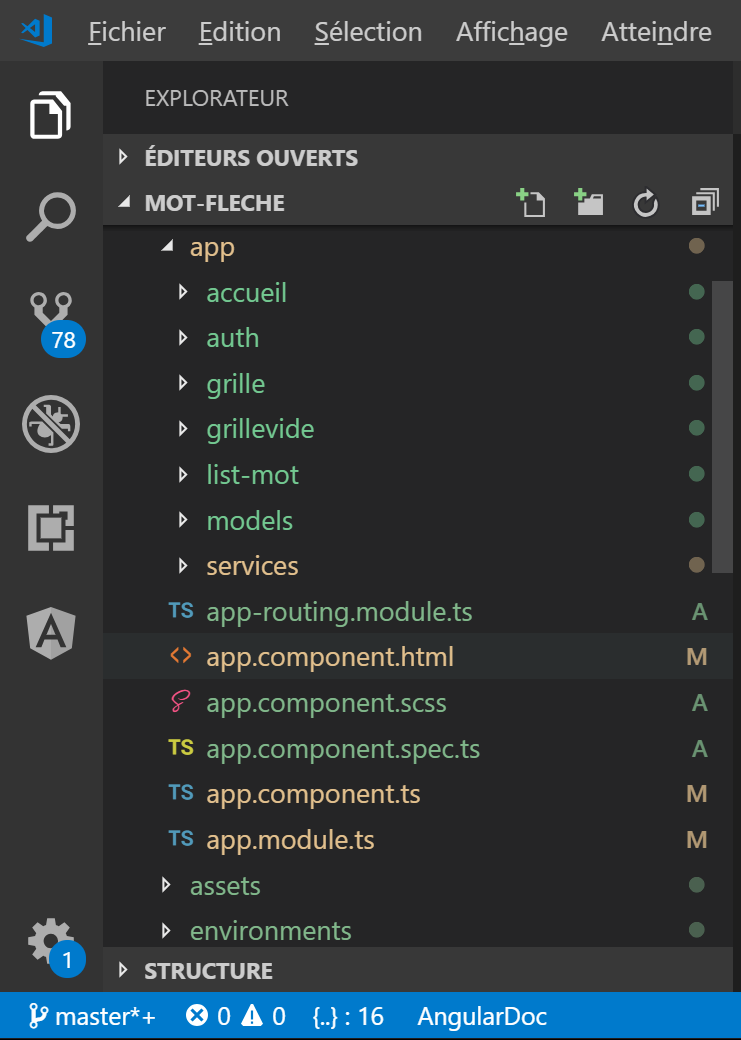
* + 1. **FireBase :**

Firebase est un ensemble de services d’hébergement pour n’importe quel type d’application (Android, iOS, Javascript, Node.js, Java….).

Il propose d’héberger en NoSQL et en temps réel des bases de données, du l’authentification social (Google, Facebook, Twitter), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.

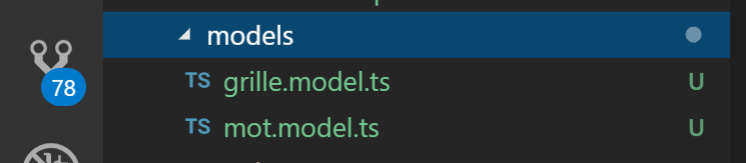
Lancé en 2011 sous le nom d’Envolve, par Andrew Lee et par James Templin, le service et racheté par Google en octobre 2014.

1. **Développement de l’application:**

Puisque Angular est un framwork basé sur le modèle MVVM et pour bien organiser le code source les différentes parties de l’application sont regroupées de la manière suivante :

* 1. **Les Composants :**

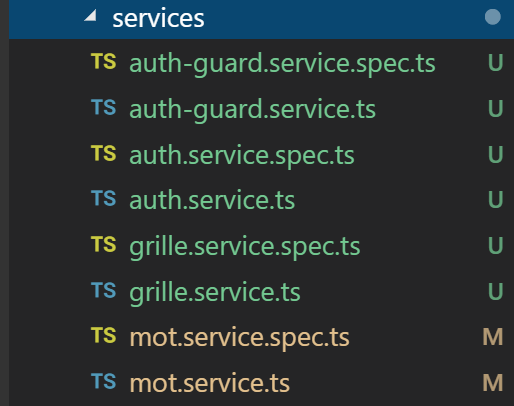
L’application contient quatre composants :

* AppComponent : c’est le composant principal : tous les autres composants de l’application seront emboités dans celui-ci
* ListMotComponent : permet d’afficher la liste des mots de la base de données et d’ajouter, supprimer ou modifier les mots.
* AccueilComponent : permet d’afficher un menu de navigation
* AuthComponent : afficher la page de connexion.
* GrilleComponent : afficher la grille avec la solution.
* GrillevideComponent : affiche une grille de mot fléchée vide et permet à l’utilisateur de jouer.
  1. ******Les Modèles :**
* Classe Mot : représente les données de la base, ses attributs sont : mot, définition.
* Classe Grille : représente la grille. Elle contient tous les méthodes qui permettent de générer une grille de mots fléchés à partir de la liste de mot enregistrer dans la base. ses attributs sont :

Tab : matrice de caractères

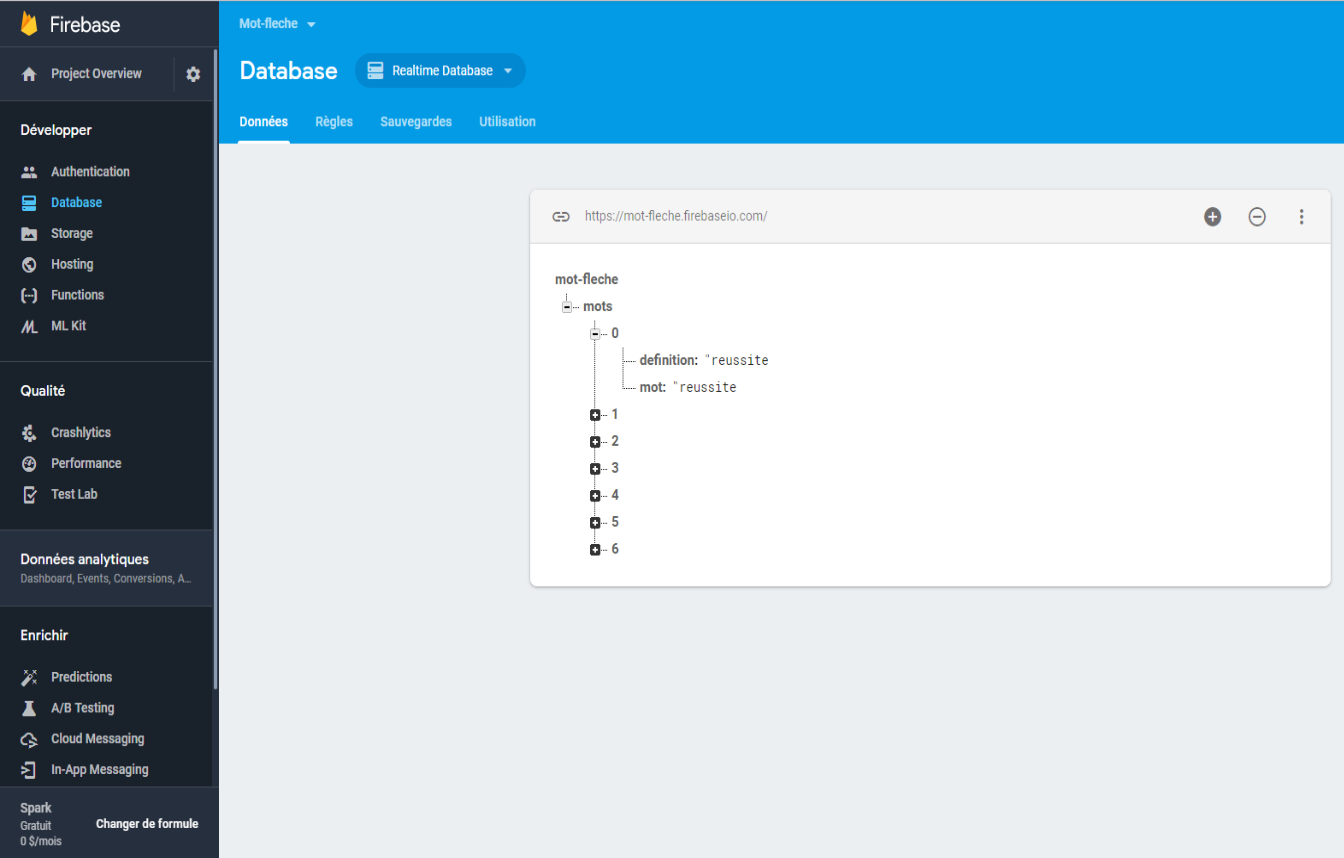
List de mot : c’est la liste de mot dans la grille

La taille de la grille.

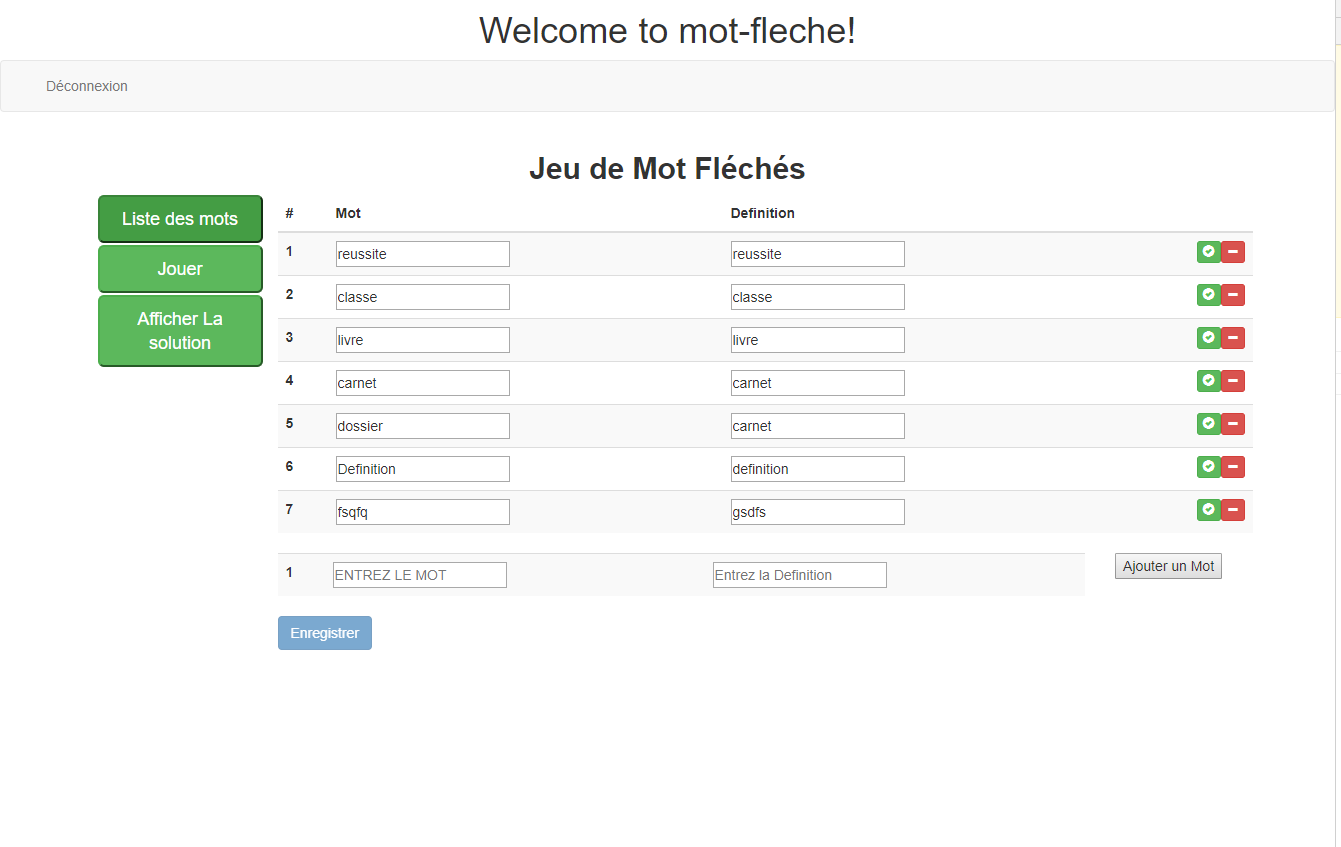
* 1. ******Les Services :**
* MotService : cette classe contient toutes les méthodes permettant de récupérer, supprimer, modifier et ajouter des mots dans la base de données.
* GrilleService : cette classe contient toutes les méthodes permettant de générer une grille de mots fléchés à partir d’une liste de mot.
* AuthService : permet de créer un compte utilisateur, de se connecter et de se déconnecter via une adresse e-mail et un mot de passe (Firebase propose différent mode de connexion (email, téléphone, Facebook, Google,..) toutes les méthodes liées à l’authentification se trouve dans firebase.auth()).
* AuthGuardService : cette classe contient une seule méthode qui permet de vérifier si l’utilisateur est authentifié. S’il est authentifié alors il peut accéder à la page d’accueil de l’application.

1. **Réalisation :**
   1. **Partie Back-end :**

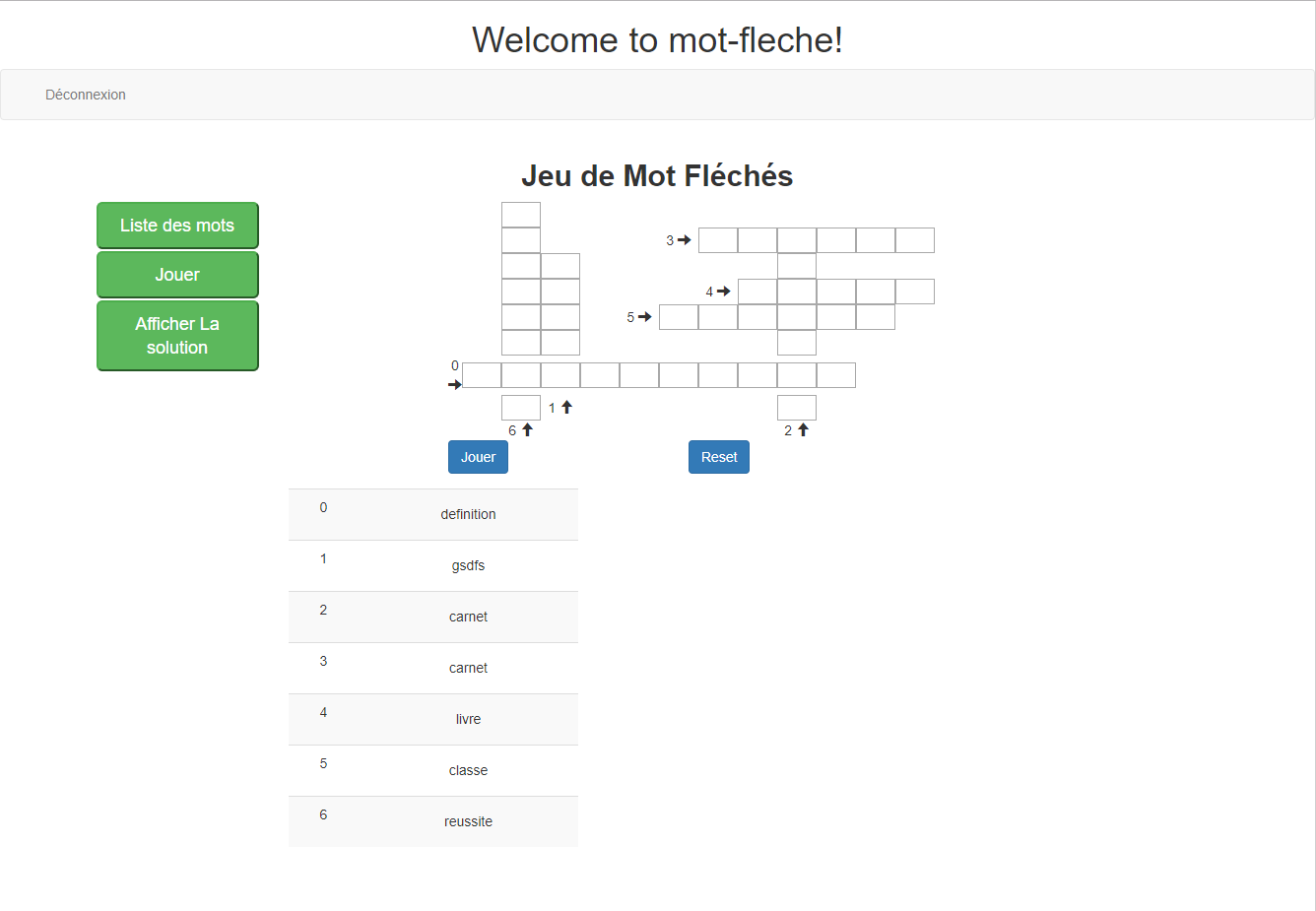
* **La liste des mots enregistrés dans la base:**

****

* 1. **Partie Front-end:**
* **La page d’accueil:**

****

* **Jouer:**

****

1. **Conclusion :**

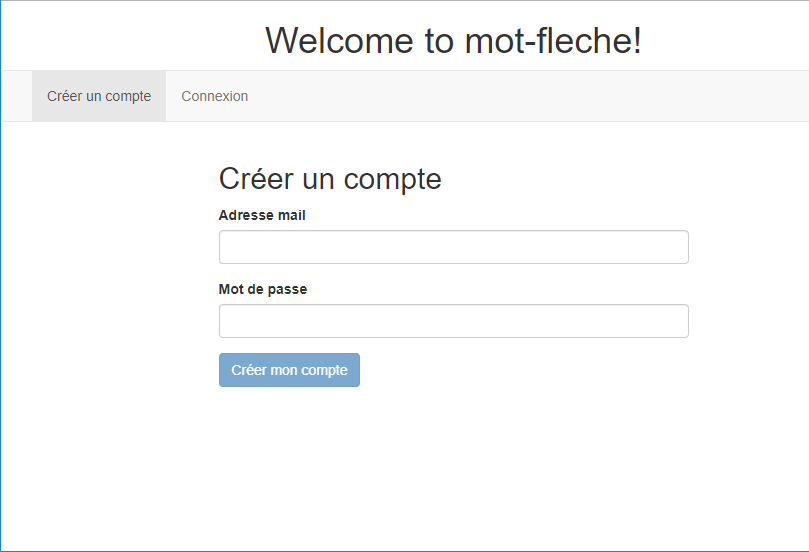
Ce stage m’est apparu comme une expérience très satisfaisante et enrichissante.

En effet le stage consistait à réaliser une application de génération de mots fléchés avec en Angular et sauvegarder les données sur une base de données Firebase. Il m’a permis d’améliorer mes connaissances en création d’application web, et notamment en ce qui concerne le respect de l’architecture logicielle. J’ai également appris l’utilisation du framework Angular avec lequel j’ai développé mon application.

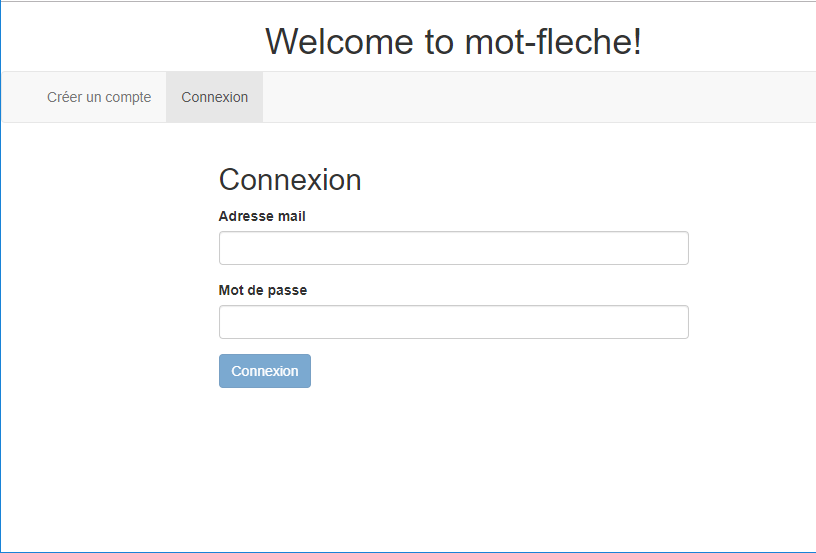
A l’heure actuelle, l’application est prête à être utilisée. On peut donc affirmer que le but qui m’avait été fixé a été atteint.

En conclusion, ce stage est complémentaire à mon année d’étude au sein de la formation Licence Informatique. Il m’a apporté une nouvelle expérience, des nouvelles connaissances professionnelles et humaines et m’a permis de comprendre les enjeux d’un projet informatique.

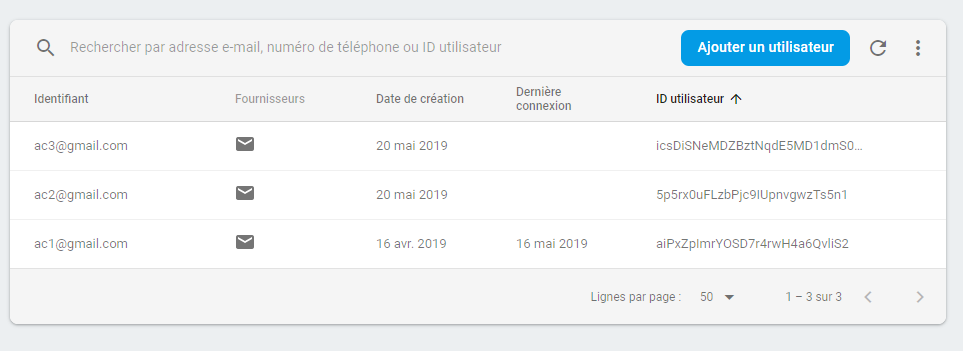
1. **Annexes :**

****

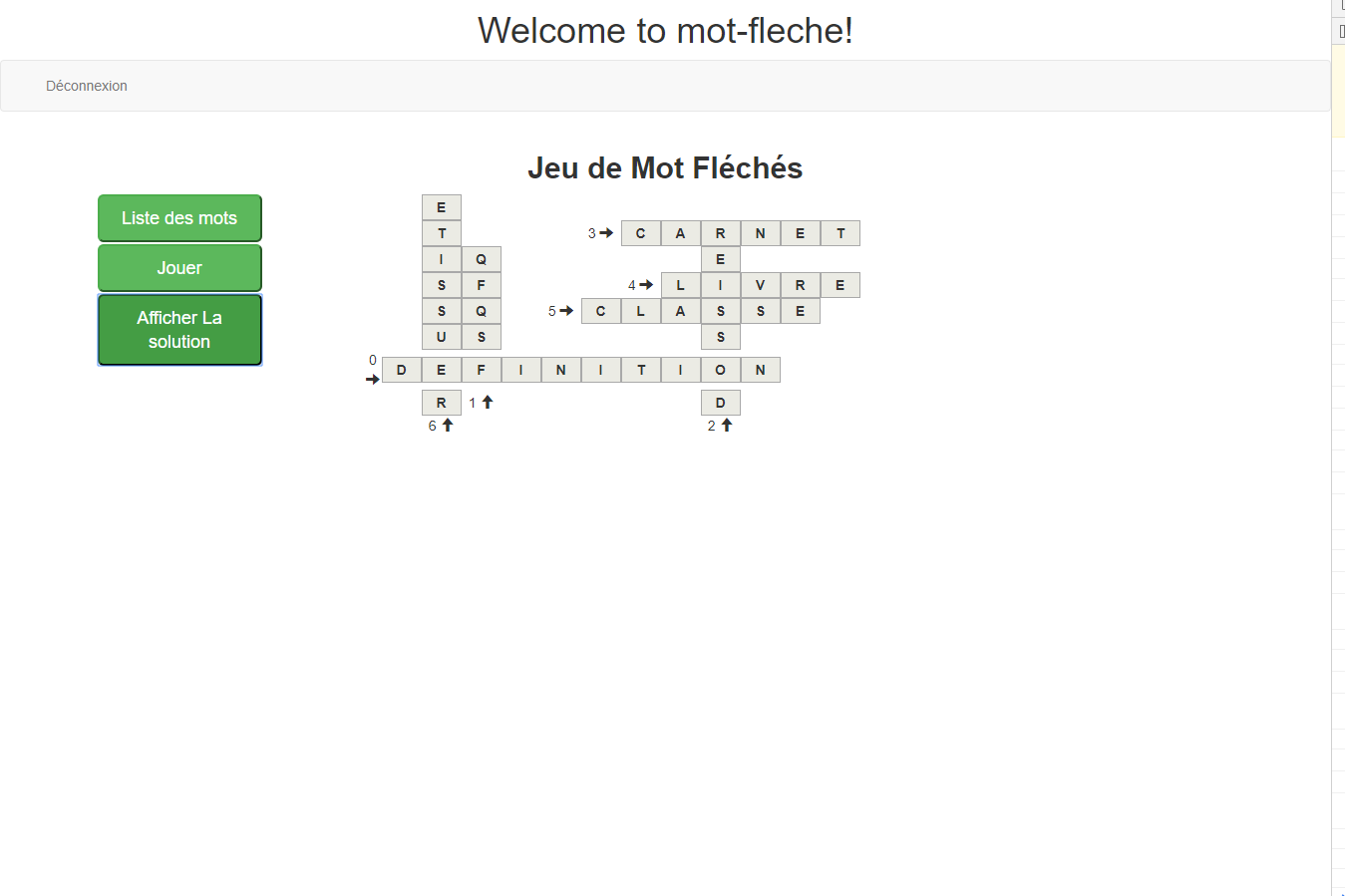
Page pour créer un compte utilisateur



Page d'authentification



Liste des utilisateurs



Affichage de la solution de la grille

1. **Glossaire :**

**Framework :** est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d’une partie d’un logiciel (architecture).

**Front-End :** désigne les éléments d’un site que l’on voit à l’écran et avec les quels on peut interagir depuis un navigateur.

**Back-End :** est un terme désignant un étage de sortie d’un logiciel devant produire un résultat. On l’oppose au front-end qui lui est la partie visible de l’iceberg.

**TypeScript :** est un langage de programmation libre et open source développé par Microsoft qui a pour but d’améliorer et de sécuriser la production de code JavaScript. C’est un sur-ensemble de JavaScript (c’est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScrpit).

**UML :** Langage de Modélisation Unifié : est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d’un système.

**NoSQL : pour** « Not only SQL », désigne les bases de données qui ne sont pas fondées sur l’architecture classique des bases de données relationnelles.

1. **Bibliographie :**

* Documentation Angular: <https://angular.io/docs>
* Tutoriel Angular: <https://openclassrooms.com/fr/courses/4668271-developpez-des-applications-web-avec-angular>
* Cours IHM :L3 Informatique Université de Rouen
* <https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal>