

A full-page illustration of Tanjiro Kamado from the anime 'Demon Slayer'. He is standing in a forest of purple wisteria trees, looking upwards with a surprised expression. He is wearing his signature blue cloud-patterned haori over a dark kimono. A sword is visible at his waist. The scene is bathed in a soft purple light.

MANGATECH

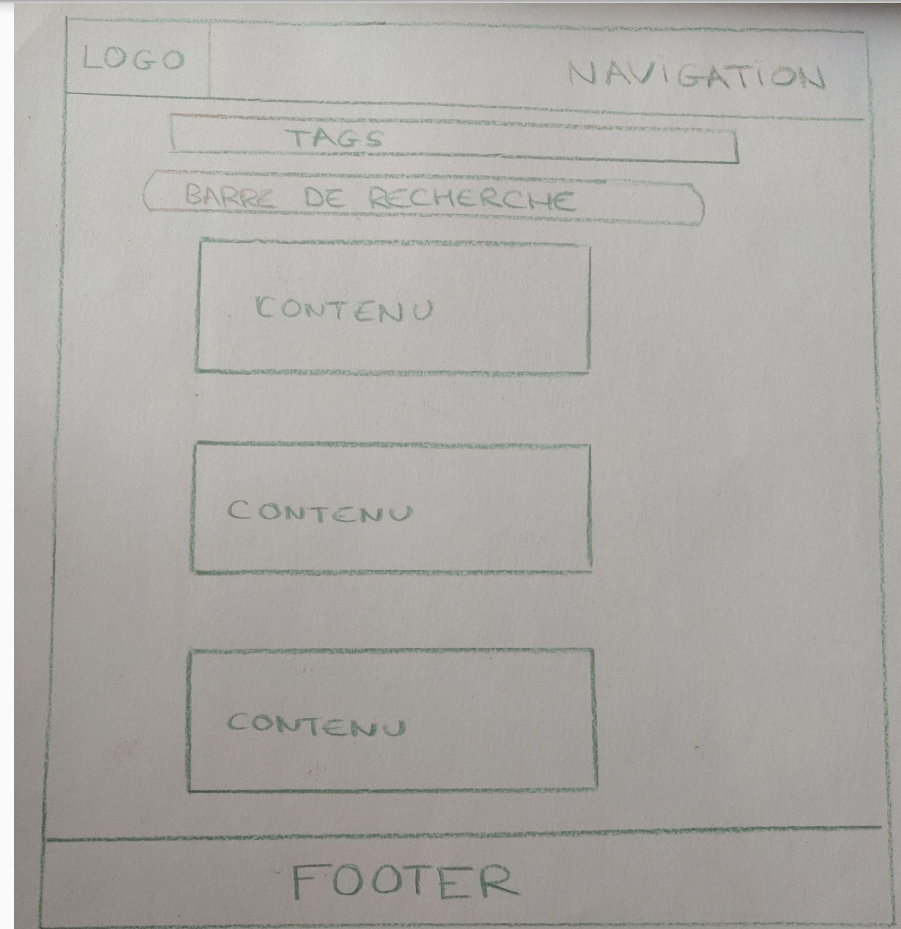
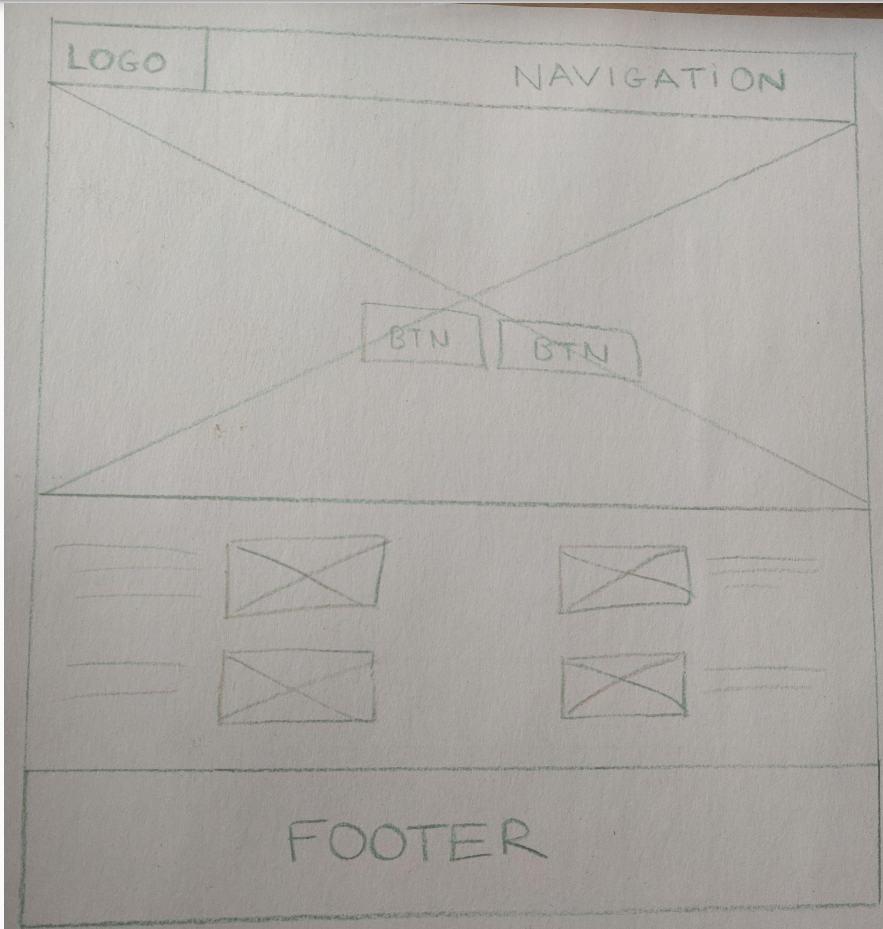
Expression des besoins

<p>Le visiteur peut :</p> <ul style="list-style-type: none">- Voir la liste des items.- Filtrer un item par tags.- Chercher un item avec une barre de recherche.- Naviguer dans le forum et filtrer les discussion par sujet.- S'inscrire.	<p>L'utilisateur connecté peut :</p> <ul style="list-style-type: none">- Ecrire un commentaire sous les items.- Ecrire une nouvelle discussion dans le forum.- Supprimer / éditer leurs propre messages.- Ajouter un item à leur liste de favori / Fini / à regarder plus tard.- Accéder à leur espace utilisateur où ils pourront consulter leurs listes.	<p>L'admin gère :</p> <ul style="list-style-type: none">- Les utilisateurs (supprimer / modifier / ajouter).- Les discussions (supprimer).- Les commentaires (supprimer).- les items (supprimer / modifier / ajouter).- les sujets (supprimer / modifier / ajouter).- les tags (supprimer / modifier / ajouter).
---	---	--

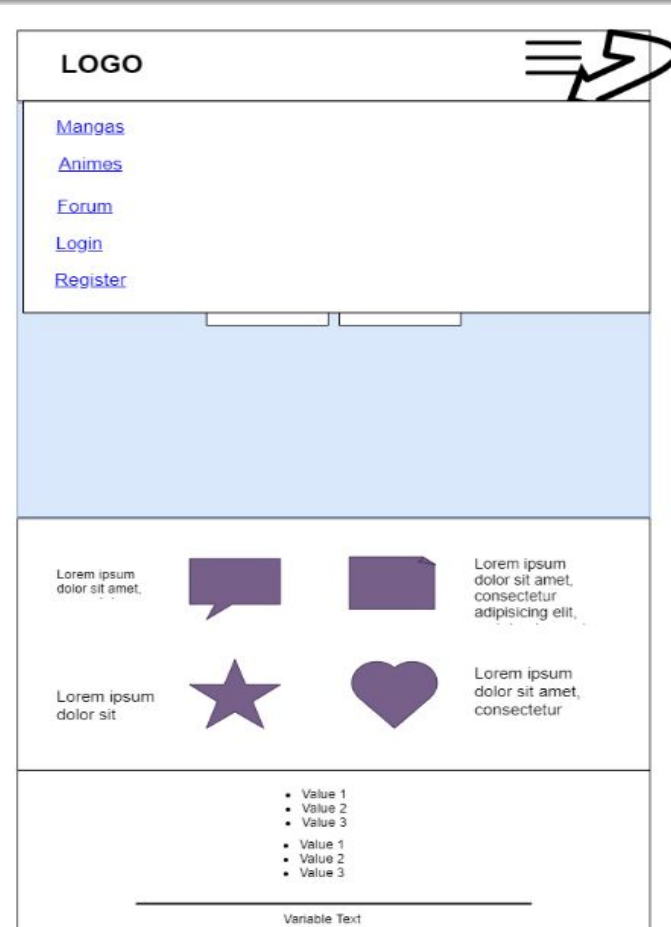
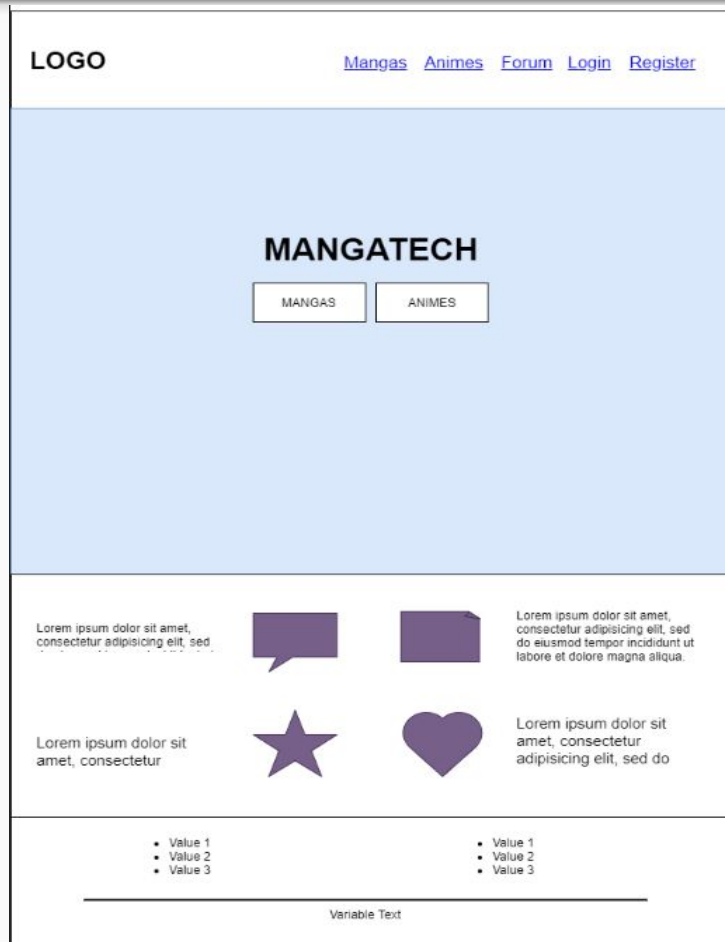
A close-up, slightly blurred photograph of a dark-colored desk. On the desk are several items: a roll of yellow tape in the top left, a pair of scissors on the right, several blue pens and pencils scattered across the surface, and several sheets of paper. Some papers feature hand-drawn geometric diagrams, including triangles and lines, while others appear to be templates or forms with text and checkboxes. The lighting is soft, creating a professional and creative atmosphere.

MODELISATION DE L'APPLICATION

MAQUETTAGE - ZONING



MAQUETTAGE - WIREFRAME



LOGO

[Mangas](#) [Animes](#) [Forum](#) [Login](#) [Register](#)

Mangas

tag

tag

tag

tag



Image

Heading

Lorem ipsum dolor sit amet, consectetur

Voir plus



Image

Heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit,

Voir plus



Image

Heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit,

Voir plus



- Value 1
- Value 2
- Value 3

- Value 1
- Value 2
- Value 3

Variable Text

LOGO



Mangas

tag

tag

tag

tag



Image

Heading

Lorem ipsum dolor sit amet, consectetur

Voir plus



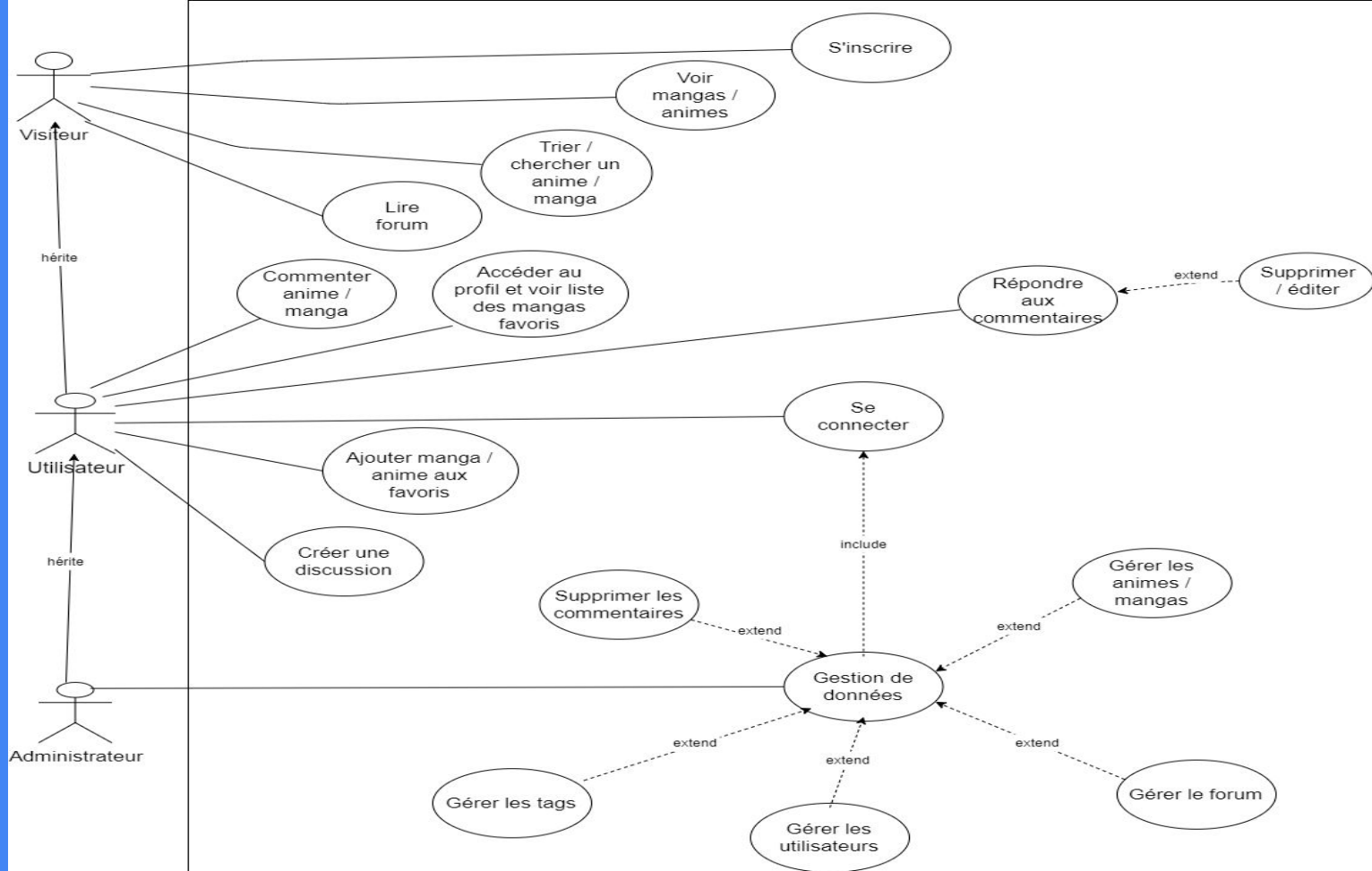
- Value 1
- Value 2
- Value 3

- Value 1
- Value 2
- Value 3

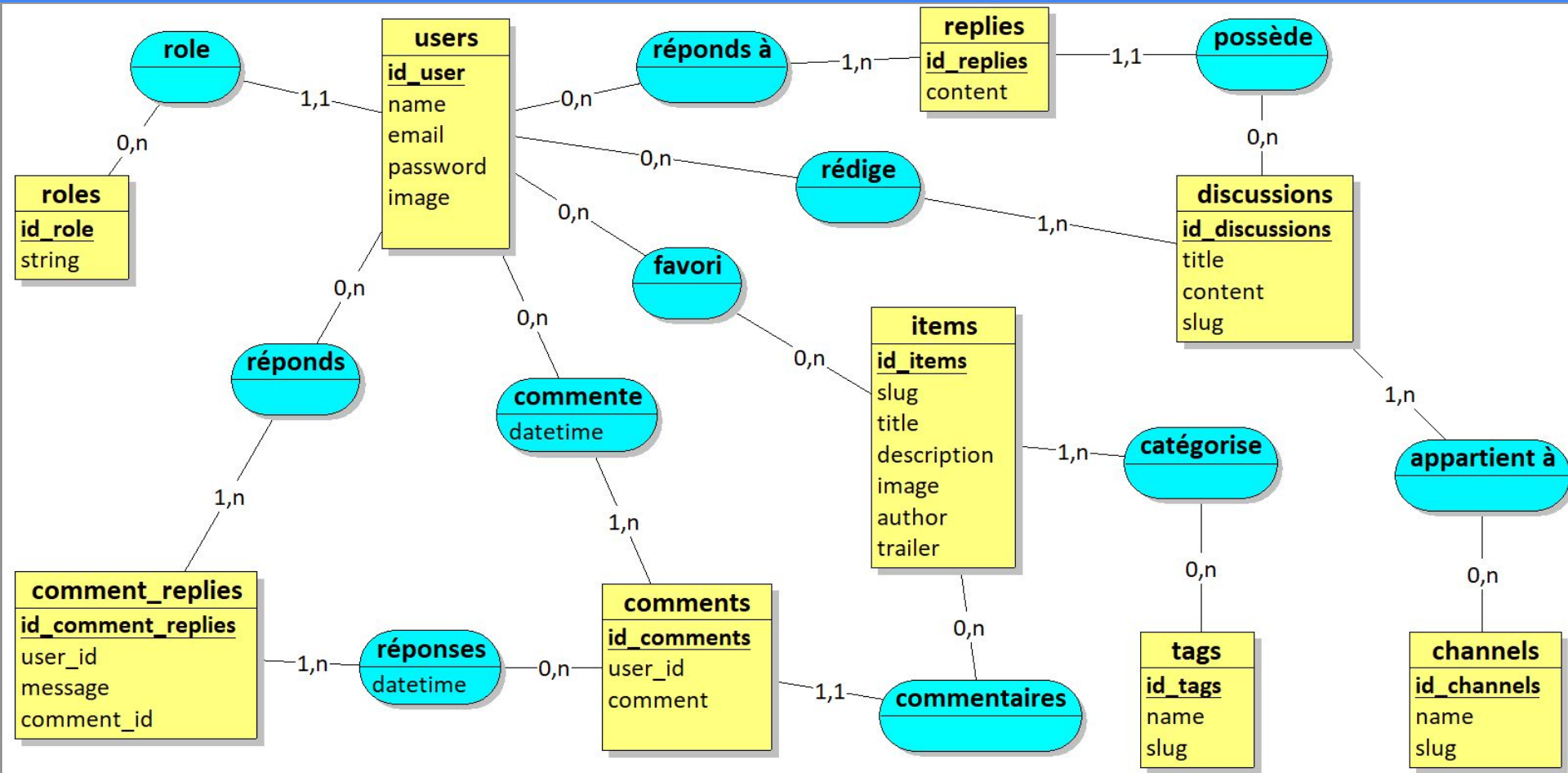
Variable Text

UML

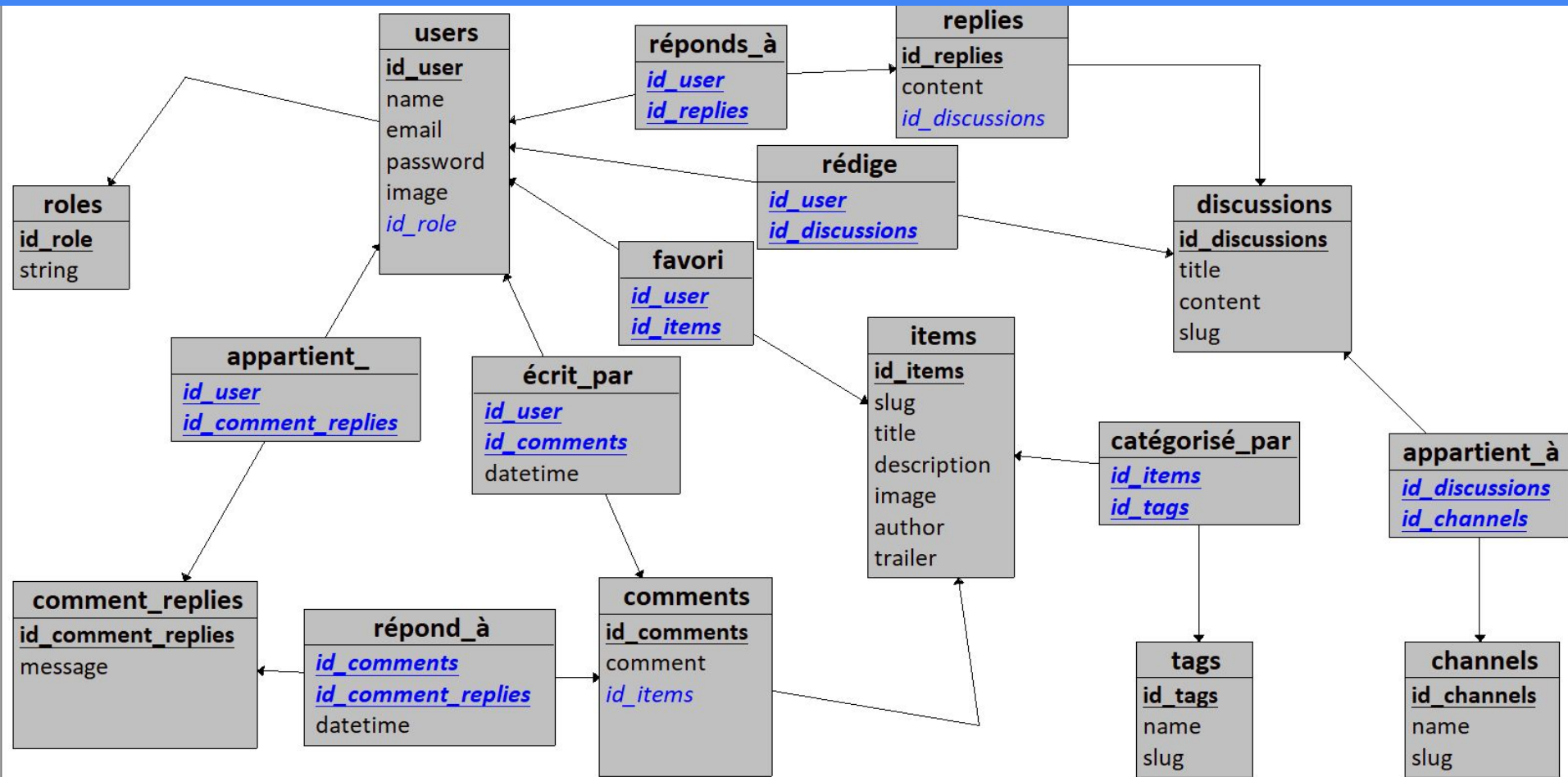
Diagramme de cas d'utilisation



MERISE - MCD



MERISE - MLD



LANGAGES UTILISÉS

HTML



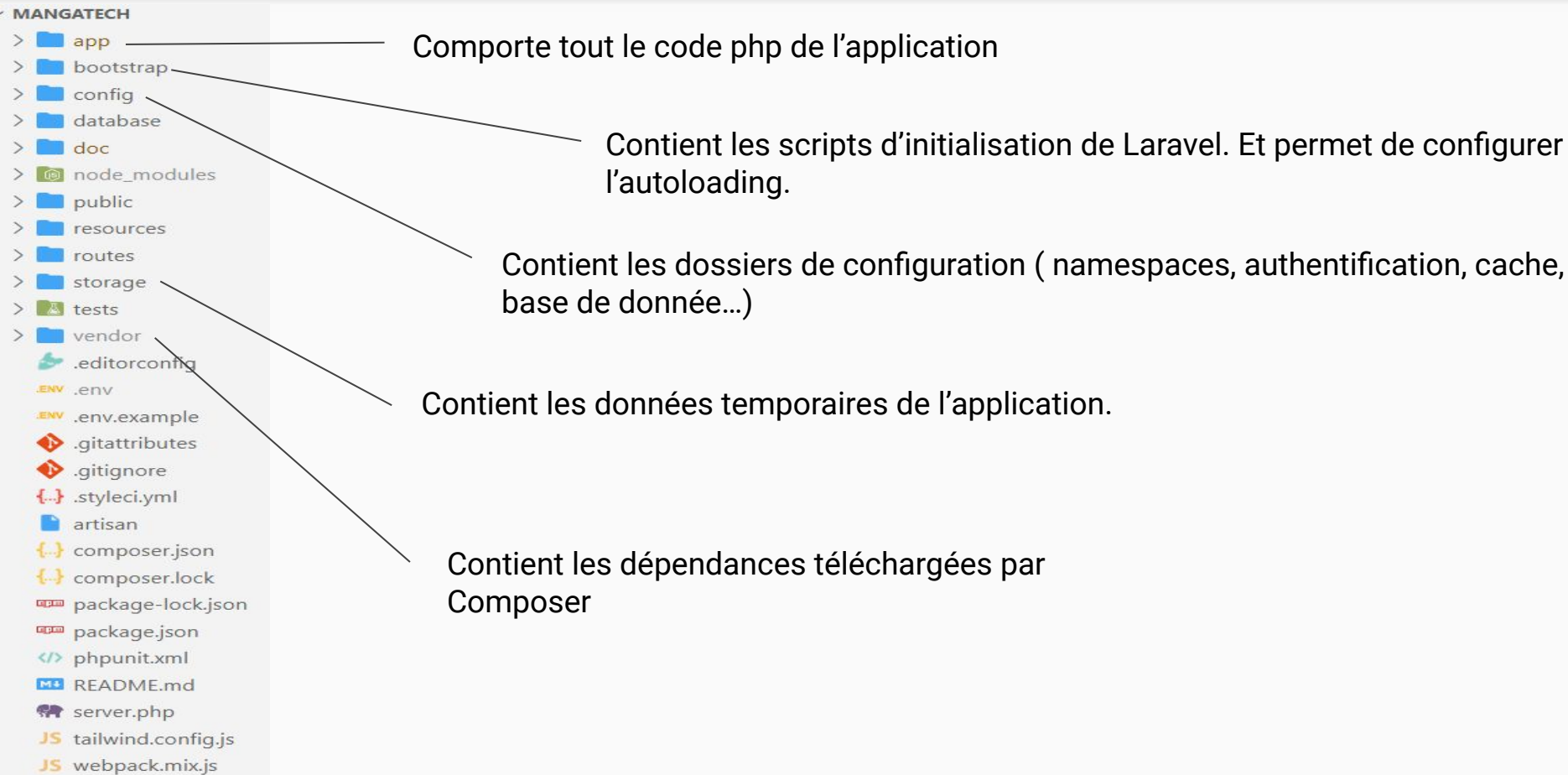
CSS



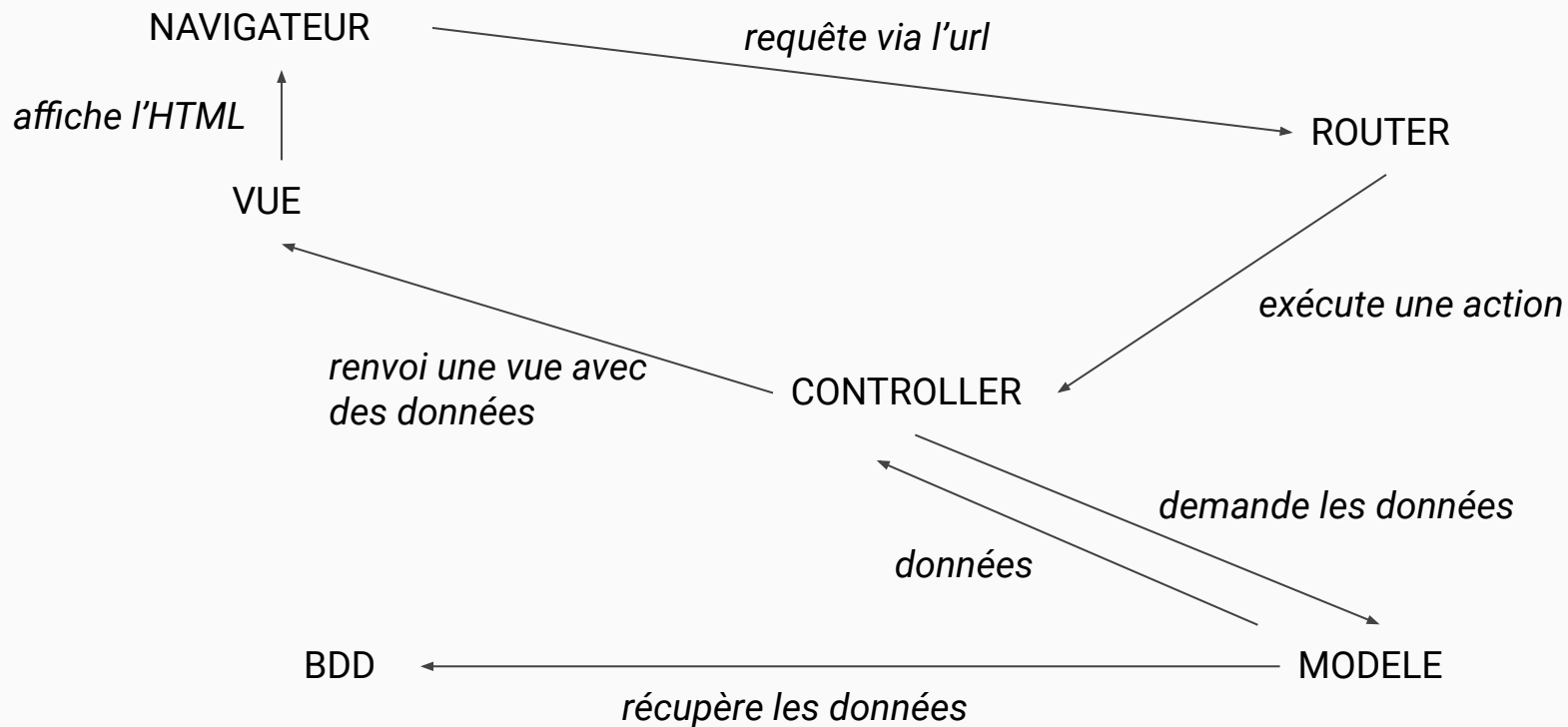
JS



STRUCTURE DU PROJET



MVC



REALISATION DE L'APPLICATION



Je crée une nouvelle base de donnée sur phpMyAdmin

J'établis la connexion de mon application à la base de donnée dans le fichier .env

```
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=mangatech
DB_USERNAME=root
DB_PASSWORD=
```

LA MIGRATION

php artisan make:migration create_discussions_table

php artisan migrate

2021_04_22_080846_create_discussions_table.php X

database > migrations > 2021_04_22_080846_create_discussions_table.php > ...

```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateDiscussionsTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('discussions', function (Blueprint $table) {
17             $table->id();
18             $table->integer('user_id');
19             $table->string('title');
20             $table->text('content');
21             $table->string('slug');
22             $table->integer('channel_id');
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::dropIfExists('discussions');
35     }
36 }
37
```

```
CREATE TABLE discussions(  
    id_discussions INT NOT NULL AUTO_INCREMENT,  
    title VARCHAR(250),  
    slug VARCHAR(250),  
    content TEXT,  
    user_id INT,  
    channel_id INT,  
    PRIMARY KEY(id_discussions),  
    FOREIGN KEY(user_id) REFERENCES users(id),  
    FOREIGN KEY(channel_id) REFERENCES channels(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

LE MODELE

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;
```

```
class Discussion extends Model
```

```
{  
    use HasFactory;  
  
    protected $fillable = [  
        'title',  
        'content',  
        'channel_id',  
        'slug',  
    ];  
  
    public function user()  
    {  
        return $this->belongsTo(User::class);  
    }  
  
    public function channel()  
    {  
        return $this->belongsTo(Channel::class);  
    }  
  
    public function replies()  
    {  
        return $this->hasMany(Reply::class);  
    }  
  
    public function getRouteKeyName()  
    {  
        return 'slug';  
    }  
}
```

php artisan make:model Discussion

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;
```

```
class Reply extends Model
```

```
{  
    use HasFactory;  
  
    protected $fillable = [  
        'content',  
        'discussion_id',  
        'user_id',  
    ];  
  
    //Reply's author  
    public function owner()  
    {  
        return $this->belongsTo(User::class, 'user_id');  
    }  
  
    public function discussion()  
    {  
        return $this->belongsTo(Discussion::class);  
    }  
}
```


LE CONTROLLER

```
class DiscussionController extends Controller
```

```
{  
    public function __construct()  
    {  
        //Si user auth ? il peut utiliser les routes create et store : affiche la login page  
        $this->middleware('auth')->only(['create', 'store']);  
    }  
}
```

```
public function index()  
{  
    return view('forum.index');  
}
```

```
public function create()  
{  
    return view('forum.discussion.create');  
}
```

```
public function store(CreateDiscussionRequest $request)  
{  
    // On prend l'user connecté et appelle sa relation hasMany, discussions(), pr lui en créer une nouvelle  
    $post = auth()->user()->discussions()->create([  
        'title' => $request->title,  
        'content' => $request->content,  
        //le slug va correspondre a la valeur du title transformé en snake_case  
        'slug' => snake_case($request->title),  
        'channel_id' => $request->channel,  
    ]);  
    $post->save();  
  
    Toastr::success('Comment has been posted', 'Success');  
    return redirect()->route('forum.index');
```

php artisan make:controller DiscussionController -r

app/Http/Controllers/Anime/CommentAnimeController

```
public function update(Request $request, $id)
{
    // on stocke la requete validée dans une variable
    $updateData = $request->validate([
        'comment' => 'required|max:1000',
    ]);
    //on valide si le commentaire correspond à l'id et on update la requete de la variable
    Comment::whereId($id)->update($updateData);

    Toastr::success('Comment has been edited', 'Success');
    return redirect()->action([AnimeController::class, 'index']);
}

public function destroy($id){
    //on cherche l'id de comment - 404 error si introuvable
    $comment = Comment::findOrFail($id);
    if ($comment->user_id == Auth::id() ) {
        //on supprime les reply qd le root comment est supprimé
        $replies = CommentReply::where('comment_id', $id)->delete();
        // SQL format : DELETE * FROM comment_replies
        //INNER JOIN comments ON comments.id = comment_replies.comment_id
        //WHERE comments.id = :id
        //on supprime le commentaire
        $comment->delete();
    }

    Toastr::success('Comment has been deleted', 'Success');
    return back();
}
```

LA VUE

view('forum.index')

variables de
structure

section dans
layouts.app

```
@include('partial.forum.sidebar')

<div class="w-full flex flex-col h-screen overflow-y-hidden">

    @include('partial.forum.header')

    <div class="w-full flex-grow p-6 overflow-auto">

        @foreach ($discussions as $discussion)

            <!--Card-->
            <div class="rounded w-full overflow-hidden shadow-lg text-center">
                <div class="px-6 py-4">
                    <div class="font-bold text-xl mb-2 text-left">{{ $discussion->user->name }}</div>
                    <span class="block text-right">{{ $discussion->created_at }}</span>
                    <div class="font-bold text-xl mb-2"><a href="{{ route('discussions.show', $discussion->slug) }}">{{ $discussion->
                    <p class="text-gray-700 text-base">
                        {{ $discussion->content }}
                    </p>
                </div>
            </div>
            <div class="px-6 pt-4 pb-2 flex justify-end">
                <span class="inline-block bg-blue-500 text-gray-100 rounded-full px-3
                py-1 text-sm font-semibold mr-2 mb-2">{{ $discussion->channel->name }}</span>
            </div>
        </div>

        @endforeach

    </div>
</div>
@endsection

@include('partial.admin.footer')
```

intégration des
variables php

```

<!doctype html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- CSRF Token -->
  <meta name="csrf-token" content="{{ csrf_token() }}">

  <title>{{ config('app.name', 'Mangatech') }}</title>

  <script src="https://kit.fontawesome.com/efbe1e177e.js" crossorigin="anonymous"></script>

  <!-- Styles -->
  <link href="{{ mix('css/app.css') }}" rel="stylesheet">
  <link rel="stylesheet" href="http://cdn.bootcss.com/toastr.js/latest/css/toastr.min.css">
</head>
<body class="{{ bg-gray-100 h-screen antialiased leading-none font-sans }}">
  <div id="app">
    <header class="{{ bg-blue-900 py-6 }}"> ← navbar
    </header>
    <main>
      @yield('content')
    </main>

    <footer>
      @include('layouts.footer')
    </footer>
  </div>
  <!-- Scripts -->
  <script src="{{ mix('js/app.js') }}" defer></script>
  <script src="http://cdn.bootcss.com/jquery/2.2.4/jquery.min.js"></script>
  <script src="http://cdn.bootcss.com/toastr.js/latest/js/toastr.min.js"></script>
  {!! Toastr::message() !!}
</body>
</html>

```

nom du fichier

@extends('layouts.app')

```

<style>
  @import url('https://fontawesome.com/css/all.css');
  .font-family-karla { font-family: karla; }
  .bg-sidebar { background-color: #3d4451; }
  .cta-btn { color: #3d4451; }
  .upgrade-btn { background-color: #3d4451; }
  .upgrade-btn:hover { background-color: #3d4451; }
  .active-nav-link { background-color: #3d4451; }
  .nav-item:hover { background-color: #3d4451; }
  .account-link:hover { background-color: #3d4451; }
</style>

```

@section('content')

haut du fichier précédant

LA ROUTE



```
1 Route::post('comment-reply/{comment}', [CommentReplyController::class, 'store']  
2 )->name('comment-reply.store');
```

Nom de la route utilisé pour
l'appeler dans la vue

méthode http

URL de la route

action du controller



```
1 Route::resource('discussions', DiscussionController::class);
```

GET HEAD	web	discussions	discussions.index	App\Http\Controllers\Forum\DiscussionController@inde
POST	web	discussions	discussions.store	App\Http\Controllers\Forum\DiscussionController@stor
	web			
GET HEAD	auth	discussions/create	discussions.create	App\Http\Controllers\Forum\DiscussionController@crea
	web			
GET HEAD	auth	discussions/{discussion}	discussions.show	App\Http\Controllers\Forum\DiscussionController@show
PUT PATCH	web	discussions/{discussion}	discussions.update	App\Http\Controllers\Forum\DiscussionController@upda
DELETE	web	discussions/{discussion}	discussions.destroy	App\Http\Controllers\Forum\DiscussionController@dest
GET HEAD	web	discussions/{discussion}/edit	discussions.edit	App\Http\Controllers\Forum\DiscussionController@edit
	web			

MERCI !