



Master Technologies de l'Hypermédia

Domaine : Sciences, Technologies, Santé

Mention : Informatique

Parcours : Technologies de l'Hypermédia (THYP)

Cahier de charges

My-WatchWise-VF

Réalisé par : Issam AISSAOUI IDRISI

Encadré par : Professeur Imad SALEH

Cours : Théories et conception des hypermédias

1. Introduction

a) Qu'est-ce qu'un cahier des charges

Un cahier des charges est un document contractuel et méthodologique qui définit de manière précise les besoins, les objectifs, les contraintes et les exigences techniques d'un projet.

Il constitue un référentiel de communication entre le porteur du projet (maître d'ouvrage) et le concepteur-réalisateur (maître d'œuvre), permettant d'assurer une compréhension commune des attentes et d'orienter le processus de conception.

Dans le domaine des technologies du web et des hypermédias, ce document ne se limite pas à la description fonctionnelle : il englobe également la réflexion sur l'expérience utilisateur, l'architecture de navigation, la modularité du système, les aspects d'interopérabilité et les stratégies de personnalisation ou de recommandation.

Ainsi, le cahier des charges joue un rôle central dans la conduite du projet : il sert à planifier, concevoir, développer et évaluer le produit final en conformité avec les besoins identifiés et les contraintes techniques.

b) Caractéristiques du cahier des charges

Un cahier des charges complet se caractérise par plusieurs éléments essentiels :

- Définir clairement les objectifs du projet et les résultats attendus.
- Décrire le contexte et les motivations à l'origine de la conception du système.
- Préciser les fonctionnalités majeures et secondaires, ainsi que leur priorité.
- Fixer les contraintes techniques, ergonomiques et temporelles.
- Établir les indicateurs de qualité et de performance, garantissant la conformité du livrable.
- Assurer la cohérence du projet entre la conception théorique, la mise en œuvre technique et la finalité hypermédia.

L'objectif est d'obtenir un document à la fois technique et conceptuel, garantissant une traçabilité complète du projet depuis sa définition jusqu'à sa validation.

c) Notre cahier des charges

Dans le cadre du cours *Théories et conception des hypermédias*, assuré par Monsieur **Imad Salah**, il a été demandé à chaque étudiant de concevoir un projet numérique complet et d'en rédiger le cahier des charges détaillé.

Le présent document concerne le projet intitulé My-WatchWise-VF, une application web interactive développée en Angular, connectée à Firebase pour la gestion de l'authentification et du stockage de données, et intégrant une API de films pour permettre la recherche, la consultation et l'enregistrement de contenus cinématographiques.

Ce projet vise à offrir une expérience utilisateur fluide et intelligente grâce à une gestion avancée des routes, des services et, à terme, à un système de recommandation personnalisé basé sur les préférences et les interactions de l'utilisateur.

Il s'inscrit dans la logique d'un hypermédia moderne, combinant interactivité, multimodalité, adaptativité et connectivité.

d) Acteurs du projet

- **Maître d'œuvre :** Professeur **Imad Salah**, encadrant du module.
- **Maître d'ouvrage :** **Issam Aissaoui Idrissi**, étudiant-développeur du projet.
- **Utilisateurs finaux :** tout internaute souhaitant rechercher, sauvegarder et suivre des films via une interface simple, connectée et personnalisée.

e) Éléments clés du projet

Élément	Détail
Durée du projet	2 mois et 3 semaines (phase de conception, développement, test)
Coût	0 € – projet académique à but pédagogique
Qualité attendue	Structure Angular modulaire, bonne ergonomie, fiabilité du système d'authentification et de stockage Firebase
Méthodologie	Approche Agile (Scrum) avec itérations fonctionnelles
Livrables	Application web Angular connectée à Firebase + documentation complète (cahier des charges et présentation)

2. Historique et contexte du projet

a) Réalité actuelle

L'univers du cinéma et des séries a connu, au cours de la dernière décennie, une transformation majeure avec la généralisation des plateformes de streaming et la dématérialisation complète des supports. Les utilisateurs se retrouvent aujourd'hui dans un écosystème dense, fragmenté, où coexistent une multitude de services tels que Netflix, Disney+, Prime Video, Canal+, ou encore Apple TV+.

Cette diversité a certes enrichi l'offre, mais elle a également complexifié l'expérience utilisateur : il devient difficile de centraliser ses films favoris, de retrouver rapidement une œuvre, ou d'obtenir des recommandations personnalisées sans être limité à l'écosystème d'une seule plateforme.

Les bases de données publiques comme IMDb ou TMDb (The Movie Database) permettent de consulter des informations complètes sur les films, mais elles ne proposent pas de véritables fonctionnalités de gestion personnelle ni d'interaction dynamique avec les utilisateurs.

C'est dans ce contexte qu'intervient My-WatchWise-VF, une application web visant à combiner les atouts des deux approches — la richesse informationnelle des bases de données cinématographiques et la souplesse d'un espace personnel connecté — afin d'offrir une expérience unifiée, personnalisable et participative.

b) Problèmes posés sur les solutions existantes

Malgré la pluralité des plateformes, plusieurs lacunes persistent :

1. **Absence de centralisation** : chaque plateforme garde ses propres données d'utilisateurs, rendant impossible une vue d'ensemble de ses films ou séries favoris.

2. **Recommandations fermées** : les suggestions sont souvent limitées à la base interne du service (Netflix recommande uniquement des œuvres présentes sur Netflix).
3. **Manque de flexibilité dans la recherche** : les filtres proposés sont souvent peu ergonomiques ou ne permettent pas une exploration intuitive du contenu.
4. **Problèmes de traçabilité personnelle** : il n'existe pas de véritable système transversal permettant de suivre ce que l'on a vu, ce que l'on souhaite voir, ou ce que l'on a aimé, indépendamment du service utilisé.
5. **Manque de transparence et d'ouverture** : les algorithmes de recommandation propriétaires sont opaques et ne permettent pas à l'utilisateur de comprendre pourquoi un film lui est proposé.

Ces constats soulignent la nécessité d'un outil plus ouvert, plus adaptable et plus orienté vers l'utilisateur, offrant un contrôle total sur son expérience cinématographique.

c) Démarche et proposition de My-WatchWise-VF

Le projet My-WatchWise-VF s'inscrit dans une démarche de convergence numérique : réunir en une seule interface web les fonctionnalités essentielles de consultation, de recherche et de gestion personnalisée de films.

Développée avec Angular, la plateforme met en œuvre :

- Une authentification sécurisée via Firebase (login, inscription, réinitialisation du mot de passe) ;
- Une intégration dynamique avec une API externe (TMDb ou équivalente) pour récupérer les données de films en temps réel ;
- Une base de données Firebase pour enregistrer les favoris, les historiques et les préférences ;
- Un système modulaire de composants et services Angular, garantissant une navigation fluide et une évolutivité du code ;
- Et, à moyen terme, un module de recommandation intelligente, fondé sur les principes du filtrage collaboratif et de la similarité de contenu, permettant de proposer à l'utilisateur des films en accord avec ses goûts et son historique de consultation.
- L'approche retenue combine les principes fondamentaux de l'hypermédia :
- Interactivité, par l'engagement direct de l'utilisateur dans la recherche et la sauvegarde ;
- Adaptabilité, par la personnalisation du contenu affiché ;
- Connectivité, par l'intégration d'API et de services tiers ;
- multimodalité, par la richesse visuelle des interfaces et l'utilisation d'éléments multimédias (affiches, bandes-annonces, notes, descriptions).

Ainsi, My-WatchWise-VF ambitionne de devenir une passerelle intelligente entre l'utilisateur et l'univers cinématographique, centrée sur l'expérience, la simplicité et la recommandation.

d) Public ciblé du site

L'application s'adresse à un large public d'utilisateurs :

- **Cinéphiles curieux** souhaitant découvrir de nouveaux films selon leurs goûts ;
- **Utilisateurs réguliers de plateformes de streaming**, désireux de centraliser leurs sélections et listes de visionnage ;

- **Étudiants ou chercheurs en cinéma**, intéressés par une base documentaire enrichie et personnalisable ;
- **Grand public**, attiré par une interface simple, fluide et accessible depuis n'importe quel navigateur.

Le site se veut universel, inclusif et adaptable à différents profils d'usagers, sans barrière de langue ni de plateforme.

En somme, My-WatchWise-VF vise à remettre l'utilisateur au centre de la navigation cinématographique, en lui donnant la capacité de contrôler, personnaliser et enrichir son propre univers de visionnage.

3. Périmètre du projet

Le projet My-WatchWise-VF consiste à concevoir et développer une application web interactive dédiée à la recherche, à la consultation et à l'enregistrement de films, avec une architecture moderne fondée sur Angular et Firebase.

L'application permettra à tout utilisateur inscrit de se connecter, de rechercher un film via une API externe, d'en consulter la fiche complète, de l'ajouter à ses favoris et, à terme, de bénéficier d'un système de recommandation personnalisée basé sur ses interactions et préférences.

Ce chapitre définit les contours du projet : ce qu'il inclut, ce qu'il exclut, ainsi que les objectifs fonctionnels et non fonctionnels visés.

a) Objectifs fonctionnels

Les objectifs fonctionnels correspondent aux principales actions que le système doit permettre à l'utilisateur d'effectuer.

Ils traduisent les fonctionnalités attendues et mesurables de l'application, à savoir :

1. Authentification sécurisée via Firebase

- Création de compte (inscription)
- Connexion (login)
- Réinitialisation de mot de passe (forgot password)
- Déconnexion

2. Recherche et affichage de films

- Intégration d'une **API de films** (par exemple TMDb)
- Recherche dynamique par titre, genre ou acteur
- Affichage détaillé : affiche, synopsis, note, durée, année, bande-annonce

3. Gestion des favoris (sauvegarde personnelle)

- Ajout et suppression d'un film dans la liste de favoris
- Vérification d'un film déjà enregistré
- Consultation de la liste complète des films sauvegardés

4. Exploration et navigation hypermédia

- Structure modulaire et fluide basée sur le routage Angular
- Possibilité d'accéder à n'importe quel composant en trois clics maximum
- Adaptation responsive sur tous les écrans (PC, tablette, smartphone)

5. Système de recommandation (module à venir)

- Proposition automatique de films similaires en fonction :
 - des genres consultés,
 - des films enregistrés,
 - des préférences d'autres utilisateurs (filtrage collaboratif).
- Classement des recommandations selon popularité et historique de clics.

6. Interface utilisateur intuitive

- Utilisation de **Bootstrap** et **CSS3** pour le style, et **jQuery** pour les animations
- Présence d'un header interactif, d'une barre de recherche, de cartes de films et de boutons d'action
- Navigation fluide entre les pages principales : *Home, Login, Sign-up, Movie Details, Favorites*

b) Objectifs non fonctionnels

Les objectifs non fonctionnels concernent la qualité, la performance et la maintenabilité de l'application.

Ils garantissent que le produit final sera utilisable, fiable et évolutif.

1. Performance et rapidité

- Temps de chargement inférieur à 3 secondes sur les pages principales.
- Optimisation des requêtes API et du cache local pour réduire la latence.

2. Sécurité et confidentialité

- Authentification Firebase sécurisée par protocoles OAuth2 et SSL.
- Stockage des données utilisateurs dans une base protégée (Firestore ou Realtime Database).
- Respect du RGPD : aucune donnée sensible non autorisée ne sera collectée.

3. Compatibilité et accessibilité

- Compatibilité assurée avec les navigateurs modernes (Chrome, Edge, Firefox, Safari).
- Interface **responsive** (grilles Bootstrap) et **accessible** (clavier, lecteur d'écran).

4. Évolutivité et modularité du code

- Architecture Angular en composants et services indépendants.
- Possibilité d'ajouter ultérieurement un moteur IA de recommandation (Python/Flask ou Node.js via API).

5. Qualité visuelle et ergonomique

- Interface épurée et cohérente, centrée sur l'affiche du film et la lisibilité du contenu.
- Hiérarchisation claire des informations (titre, description, actions).

6. Traçabilité et maintenance

- Suivi du développement via GitHub et Trello.
- Documentation intégrée (commentaires TypeScript + guide utilisateur).

c) Fonctions principales du système

Fonction	Description	Utilisateur concerné
S'authentifier	Connexion à un compte utilisateur via Firebase	Utilisateur enregistré
S'inscrire	Création d'un nouveau compte sécurisé	Nouvel utilisateur
Réinitialiser le mot de passe	Envoi d'un e-mail de réinitialisation via Firebase	Utilisateur
Rechercher un film	Recherche dynamique à partir de mots-clés	Tous utilisateurs
Afficher les détails d'un film	Visualiser description, note, affiche et lien bande-annonce	Tous utilisateurs
Enregistrer un film	Ajouter un film dans la liste personnelle de favoris	Utilisateur connecté
Afficher les favoris	Consulter les films enregistrés	Utilisateur connecté
Recevoir des recommandations	Voir des propositions de films similaires	Utilisateur connecté
Se déconnecter	Quitter sa session en toute sécurité	Utilisateur connecté

d) Hors périmètre

Afin de recentrer le développement sur les fonctionnalités essentielles, certaines options sont volontairement exclues à ce stade :

- Visionnage direct de films (intégration de plateformes de streaming).
- Fonctionnalités sociales avancées (commentaires, notation, partage).
- Analyse comportementale complexe (machine learning ou IA complète).
- Version mobile native (sera envisagée dans une version ultérieure).

Le périmètre du projet s'appuie ainsi sur une approche claire, cohérente et progressive :

- **Phase 1** : Authentification et recherche.
- **Phase 2** : Gestion et affichage des favoris.
- **Phase 3** : Recommandation intelligente et personnalisation.

Ce cadrage garantit une livraison fonctionnelle rapide tout en préparant les bases d'une évolution future vers une plateforme de recommandation complète.

4. Risques et craintes

Comme tout projet numérique complexe, My-WatchWise-VF est exposé à un ensemble de risques techniques, organisationnels et fonctionnels susceptibles d'affecter sa qualité, son calendrier ou son bon fonctionnement.

L'identification précoce de ces risques permet d'anticiper les difficultés, de définir des stratégies de mitigation adaptées et de garantir une conduite de projet maîtrisée.

a) Risques techniques

1. Problèmes d'intégration avec Firebase

- *Risque* : erreurs d'authentification, limitations de quota ou mauvaise configuration des règles de sécurité Firestore.
- *Impact* : blocage de la connexion utilisateur ou perte temporaire de données.
- *Mesures de prévention* : validation des règles de lecture/écriture, tests unitaires réguliers, utilisation de l'émulateur Firebase pour le développement local.

2. Dépendance aux API externes (ex. TMDb)

- *Risque* : indisponibilité temporaire ou modification des endpoints de l'API.
- *Impact* : impossibilité d'afficher ou de rechercher des films.
- *Mesures* : implémentation d'un système de cache local (IndexedDB) et gestion des erreurs HTTP par un service Angular centralisé.

3. Problèmes de performance ou de charge

- *Risque* : lenteur des requêtes API ou surcharge du front-end.
- *Impact* : dégradation de l'expérience utilisateur.
- *Mesures* : pagination des résultats, lazy-loading des modules Angular, compression des images et audit Lighthouse.

4. Incompatibilités entre versions Angular et dépendances

- *Risque* : conflits entre Angular, Bootstrap et jQuery lors de mises à jour.
- *Impact* : comportements visuels inattendus ou erreurs de build.
- *Mesures* : verrouillage des versions stables (package-lock.json) et documentation technique à jour.

5. Sécurité et protection des données

- *Risque* : accès non autorisé ou injection de données malveillantes.
- *Impact* : atteinte à la confidentialité des utilisateurs.
- *Mesures* : filtrage côté client et côté serveur, usage obligatoire de HTTPS, authentification Firebase basée sur tokens JWT, validation stricte des entrées.

b) Risques d'usage et d'ergonomie

1. Manque d'adoption initiale

- *Risque* : difficulté à attirer les premiers utilisateurs sans réseau social intégré.

- *Impact* : faible taux d'utilisation et absence de retours.
- *Mesures* : démonstrations ciblées, valorisation visuelle, intégration d'un tutoriel interactif dès la première connexion.

2. Surcharge cognitive

- *Risque* : interface jugée complexe ou trop riche en options.
- *Impact* : désorientation ou abandon de l'utilisateur.
- *Mesures* : hiérarchisation claire des informations, test d'ergonomie avec utilisateurs, simplification progressive des menus.

3. Accessibilité limitée

- *Risque* : application difficile à utiliser pour des publics spécifiques (daltoniens, malvoyants).
- *Impact* : exclusion d'une partie du public.
- *Mesures* : respect des normes **WCAG 2.1**, contrastes adaptés, ajout d'une recherche vocale et d'un mode clair/sombre.

4. Problème de confiance dans la recommandation

- *Risque* : si les recommandations semblent aléatoires ou peu pertinentes, l'utilisateur peut douter de leur fiabilité.
- *Mesures* : transparence sur les critères utilisés, amélioration continue du moteur de recommandation par analyse des retours utilisateurs.

c) Risques organisationnels et temporels

1. Sous-estimation du temps de développement

- *Risque* : accumulation de retards entre les phases d'intégration et de tests.
- *Impact* : non-respect des délais académiques.
- *Mesures* : planification agile par sprints hebdomadaires, priorisation des tâches critiques et suivi avec Trello.

2. Complexité du système de recommandation

- *Risque* : difficultés techniques à implémenter un algorithme de filtrage personnalisé dans le temps imparti.
- *Impact* : fonctionnalité partielle ou reportée.
- *Mesures* : déploiement initial d'un modèle simple (par genre et popularité), suivi d'une amélioration progressive.

3. Pannes ou pertes de données

- *Risque* : défaillance technique du service Firebase ou erreur de manipulation.
- *Mesures* : sauvegardes automatiques et export hebdomadaire des données utilisateurs.

4. Manque de coordination ou de documentation

- *Risque* : perte de cohérence dans le code et la structure du projet.
- *Mesures* : convention de nommage stricte, documentation des modules Angular et commits réguliers sur GitHub.

d) Stratégies de mitigation

Pour faire face à ces risques, plusieurs stratégies globales sont adoptées :

- **Agilité et flexibilité** : adaptation continue du plan de travail en fonction des obstacles rencontrés.
- **Documentation vivante** : mise à jour du cahier des charges et du guide technique à chaque sprint.
- **Tests réguliers** : validation de chaque fonctionnalité dès sa livraison.

- **Sauvegarde et supervision** : surveillance de l'application et duplication régulière de la base de données.
- **Formation continue** : auto-formation sur Angular et Firebase via documentation officielle et tutoriels, afin de maîtriser rapidement les nouveautés.

En anticipant ces risques, le projet **My-WatchWise-VF** s'inscrit dans une logique de **sécurisation du cycle de développement**, garantissant une application stable, évolutive et conforme aux attentes des utilisateurs.

5. Missions et découpage des tâches

Le projet **My-WatchWise-VF** a été conçu selon une méthodologie **Agile simplifiée**, inspirée du cadre **Scrum**, adaptée à une durée de **3 semaines**.

L'objectif est de livrer un prototype fonctionnel complet, en privilégiant l'efficacité, la clarté du code et la cohérence ergonomique.

Chaque semaine correspond à un **micro-sprint**, incluant conception, développement, validation et documentation.

a) Méthodologie retenue

La méthode Agile permet de structurer un projet court tout en conservant une logique incrémentale. Chaque sprint aboutit à un livrable concret testable, garantissant une progression visible et mesurable :

- **Sprint 1 : Conception et mise en place de l'environnement.**
- **Sprint 2 : Développement des fonctionnalités principales (authentification, recherche, affichage).**
- **Sprint 3 : Intégration des favoris, optimisation et finalisation.**

À chaque fin de sprint, une **revue du travail accompli** est effectuée pour vérifier la conformité aux objectifs et ajuster les priorités.

b) Acteurs du projet

Rôle	Nom / Fonction	Responsabilités principales
Product Owner	<i>Professeur Imad Salah</i>	Définit la vision du projet, oriente les choix fonctionnels, valide les livrables.
Scrum Master & Développeur	<i>Issam Aissaoui Idrissi</i>	Planifie les tâches, développe les modules Angular, intègre Firebase et l'API de films, assure les tests et la documentation.

c) Découpage des tâches principales (3 semaines)

Tâche / Sprint	Contenu	Livrable attendu
Semaine 1 – Conception & initialisation	<ul style="list-style-type: none"> - Définition du périmètre fonctionnel et des interfaces. - Installation et configuration d'Angular, Firebase et GitHub. - Création de l'arborescence du projet. - Conception du design global (Bootstrap + maquettes). 	Application initialisée + maquette du site (home, login, signup).
Semaine 2 – Développement principal	<ul style="list-style-type: none"> - Intégration de Firebase Auth (login, signup, reset). - Création des composants <i>home</i>, <i>search-bar</i>, <i>movie-details</i>. - Connexion à l'API de films (TMDb ou équivalente). - Routage dynamique et affichage des résultats. 	Application opérationnelle avec authentification et recherche.
Semaine 3 – Finalisation & tests	<ul style="list-style-type: none"> - Ajout du module <i>favoris</i> (enregistrement Firebase). - Amélioration du header et de la navigation. - Tests unitaires et intégration. - Nettoyage du code et préparation de la soutenance. 	Prototype fonctionnel final + documentation complète.

d) Outils utilisés

- **Environnement de développement :** VS Code
- **Framework front-end :** Angular
- **Base de données et Authentification :** Firebase
- **API externe :** TMDb (The Movie Database)
- **Gestion du code :** Git & GitHub
- **Design & UI :** Bootstrap / jQuery / CSS3
- **Suivi de projet :** Trello

e) Livrables attendus

1. **Prototype complet** de l'application web (Angular + Firebase).
2. **Documentation technique et académique** (présent cahier des charges + guide utilisateur).
3. **Présentation orale** illustrant les fonctionnalités et les choix technologiques.

f) Synthèse du planning global

Semaine	Phase principale	Objectif clé
Semaine 1	Conception et initialisation	Architecture Angular + Firebase prête
Semaine 2	Développement fonctionnel	Recherche et affichage des films opérationnels
Semaine 3	Intégration finale et tests	Application stable, favoris fonctionnels, documentation livrée

Cette organisation condensée sur trois semaines permet de livrer une version fonctionnelle, fluide et fidèle aux objectifs initiaux, tout en respectant le format académique du Master THYP et les contraintes temporelles du projet court.

6. Étapes et tâches du projet

Le développement de My-WatchWise-VF suit une approche structurée en trois phases : conception, développement et validation / exploitation.

Chaque phase vise à assurer la cohérence du produit, la clarté du code et la conformité avec le cahier des charges.

a) Phase 1 – Conception

Objectif : poser les bases conceptuelles et techniques du projet.

- Définition du besoin et du public cible.
- Choix de l'architecture modulaire Angular (composants + services).
- Configuration de Firebase (authentification, base, hébergement).
- Conception du design global avec Bootstrap et jQuery.
- Création des maquettes : *Home, Login, Movie-Details, Favorites*.

Livrables :

- Cahier des charges validé.
- Maquettes et arborescence.
- Environnement Angular et Firebase opérationnel.

b) Phase 2 – Développement

Objectif : implémenter les fonctionnalités principales.

1. **Authentification Firebase :**
Composants login, sign-up, forgot-password.
Gestion d'erreurs, redirections et stockage sécurisé.
2. **Recherche de films via API :**
Service movie-api.service.ts, composant search-bar.
Requêtes TMDb, affichage dynamique des résultats (affiche, note, synopsis).

3. **Détails et navigation :**
Composant movie-details, routage paramétré /movie/:id.
4. **Favoris Firebase :**
Service firebase-favorites.service.ts, ajout / suppression / vérification.
Composant movie-saved pour affichage des favoris.
5. **Interface responsive :**
Header global, thèmes Bootstrap, compatibilité multi-écran.

Livrables :

- Application Angular fonctionnelle.
- Intégration API + Firebase.
- Navigation fluide et design finalisé.

c) Phase 3 – Validation et exploitation

Objectif : garantir la qualité, la fiabilité et la mise à disposition du produit.

1. **Tests** : unitaires (Angular), intégration (Firebase + API), tests utilisateurs.
2. **Corrections / optimisation** : lazy-loading, cache local, nettoyage du code.
3. **Documentation et soutenance** : guide technique, démonstration orale.
4. **Mise en ligne** : déploiement Firebase Hosting.

Livrables :

- Prototype stable.
- Documentation complète.
- Application hébergée.

7. Caractéristiques du site

a) Site hypertexte

- L'ensemble des pages (Home, Login, Movie-Details, etc.) sont liées par des routes Angular (RouterModule).
- Chaque film est un nœud relié à d'autres via des liens : genre, réalisateur, popularité.
- Navigation en trois clics maximum vers toute ressource.

b) Site multimédia

- Présence de textes, images, affiches HD, bandes-annonces (iframe YouTube).
- Utilisation d'icônes FontAwesome et d'animations CSS.
- Fiches films enrichies : synopsis, casting, bande-son, visuels promotionnels.

c) Site adaptatif

- Design responsive grâce à Bootstrap Grid.
- Adaptation automatique au terminal (smartphone, tablette, desktop).
- Détection de thème (clair / sombre).
- Possibilité d'évolution vers personnalisation selon préférences Firebase.

d) Système de recommandation

- Recommandation par **popularité** : classement selon nombre de clics.
- Recommandation par **genre** : similarité thématique.
- Recommandation par **historique personnel** : favoris récents.
- Recommandation future (IA) : filtrage collaboratif avec apprentissage des préférences.

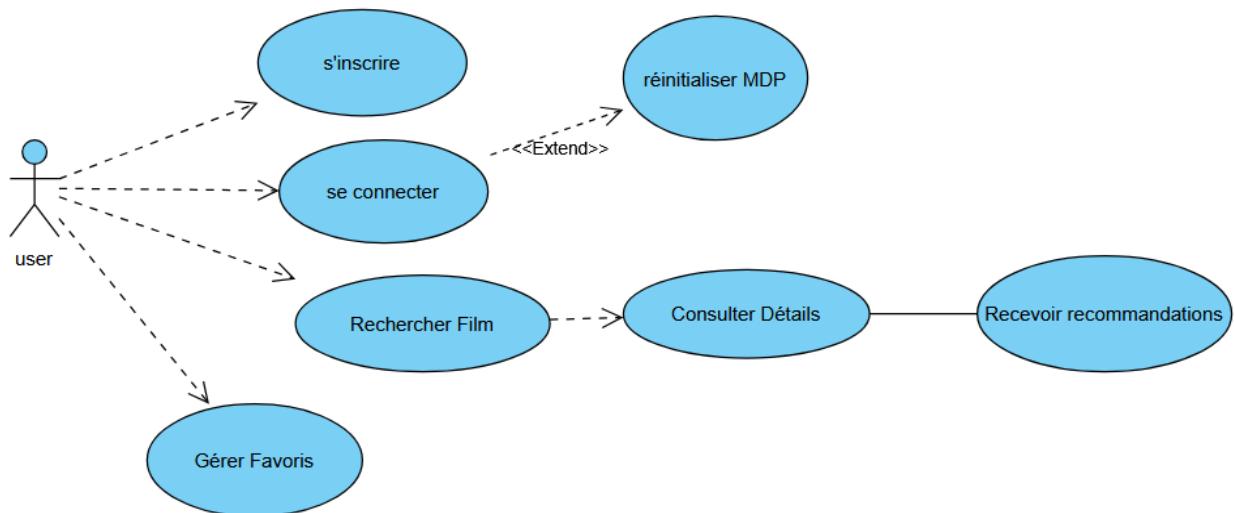
e) Interface Homme-Machine (IHM)

- Navigation intuitive et esthétique.
- Zoning clair : barre de recherche, affiches, actions rapides.
- Éléments cohérents : couleurs neutres, boutons arrondis, feedback visuel.
- Accessibilité : navigation clavier, textes alternatifs, contraste suffisant.
- Fluidité des transitions (Angular Animations).

8. UML et modélisation

La modélisation UML permet de représenter les interactions entre acteurs et systèmes, ainsi que les données gérées par Firebase.

a) Diagramme de cas d'utilisation (Use Case)

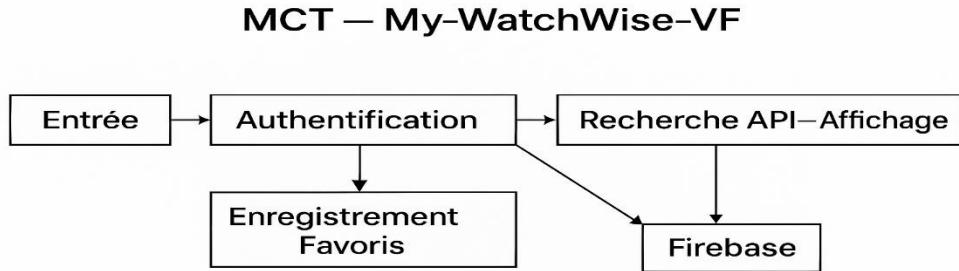


Acteurs principaux :

- Utilisateur
- Système Firebase / API TMDb
- Administrateur (validation et supervision, optionnel)

b) MCT – Modèle Conceptuel de Traitement

Processus simplifié :



Chaque traitement correspond à une fonction Angular :

- AuthService → connexion et sécurité
- Movie ApiService → communication API
- FavoritesService → gestion Firebase

c) MCD – Modèle Conceptuel de Données (Firebase)

```
Collection Users {  
    uid: String  
    email: String  
    favoris: [idFilm]  
}
```

```
Collection Films {  
    idFilm: String  
    titre: String  
    genre: String  
    note: Float  
    posterUrl: String  
}
```

Les collections Firebase sont non-relationnelles : chaque document contient ses propres clés et sous-objets, garantissant la rapidité et la scalabilité.

9. Choix techniques

Domaine	Outil / Technologie	Justification
Framework	Angular 17	Modularité, routing intégré, compatibilité Firebase
Base de données	Firebase Firestore	Temps réel, sécurité gérée par Google
Langages	TypeScript, HTML5, CSS3	Standard Angular
API	TMDb (The Movie Database)	Données filmographiques ouvertes
Design	Bootstrap 5, jQuery	Rapidité, responsive, animations
Gestion du code	GitHub	Sauvegarde, versioning
IDE	Visual Studio Code	Support Angular, extensions Firebase
Hébergement	Firebase Hosting	Déploiement simple, HTTPS intégré
Suivi de projet	Trello	Méthodologie Agile, suivi des sprints



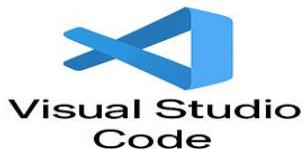
Angular 17



Firebase Firestore



Firebase Hosting



Visual Studio Code

Ces choix garantissent un équilibre entre performance, simplicité de maintenance et cohérence académique.

10. Objectifs et espérances

Les objectifs du projet sont :

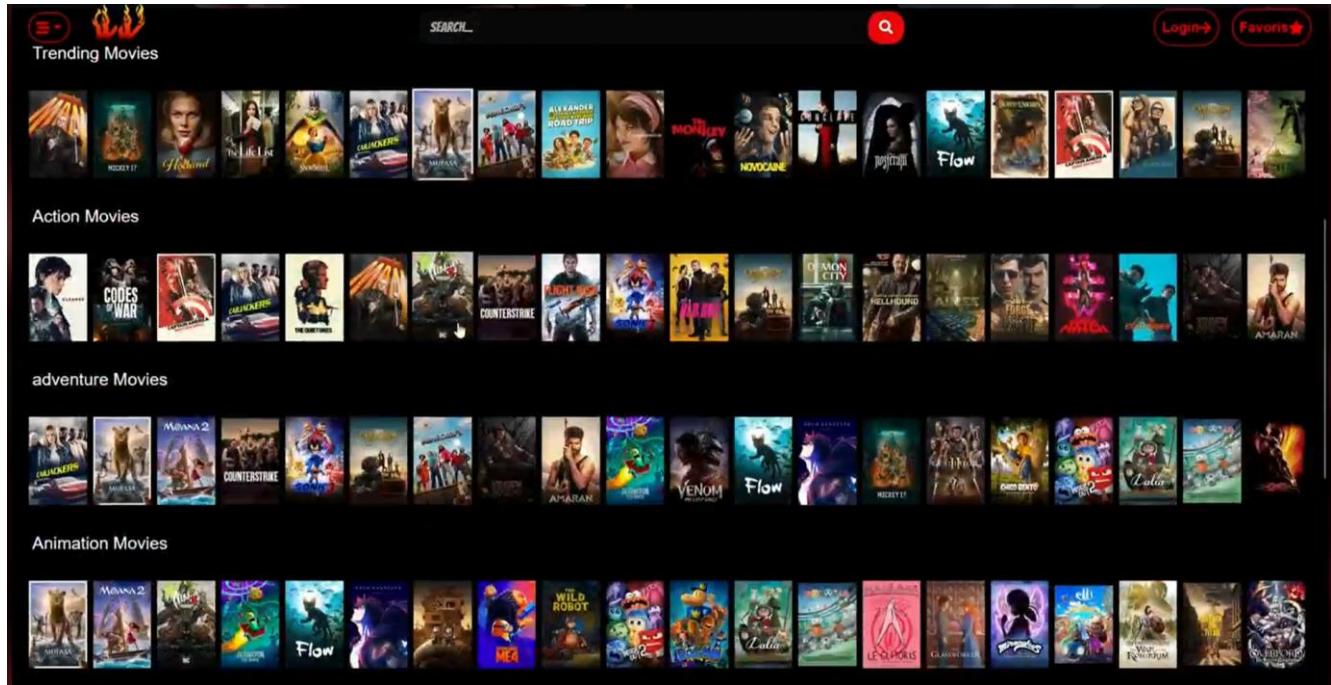
- Offrir une expérience utilisateur fluide et intuitive.
- Démontrer la maîtrise d'un framework moderne (Angular).
- Intégrer Firebase pour l'authentification et la persistance des données.
- Mettre en place un prototype de recommandation évolutif.
- Respecter le triptyque Qualité – Délai – Ressources.

Espérances :

- Déployer un produit stable, réutilisable et évolutif.

- Servir de base à une future application complète avec IA de recommandation.
- Illustrer la capacité à conduire un projet complet dans un délai restreint.

11. Réalisation



a) Structure de l'application

- *AppComponent* : racine du projet.
- *LoginComponent*, *SignUpComponent*, *ForgotPasswordComponent* : gestion utilisateur.
- *HomeComponent*, *SearchBarComponent*, *MovieDetailsComponent*, *MovieSavedComponent* : logique fonctionnelle.
- *Services* : AuthService, Movie ApiService, FavoritesService.

b) Exemple de navigation

Home → Search → Movie-Details → Add Favorite → Favorites

c) Aperçu fonctionnel

- **Accueil** : liste des films populaires.
- **Recherche** : requête API en direct.
- **Détails** : fiche film complète.
- **Favoris** : affichage personnalisé.
- **Authentification** : création / connexion Firebase.

d) Validation technique

- Tests de connexion / inscription : OK.
- Tests de recherche (API TMDb) : OK.

- Sauvegarde / suppression favoris : OK.
- Routage et redirections : OK.

12. Système de recommandation

a) Principe général

Le module de recommandation vise à suggérer des films pertinents à chaque utilisateur selon son historique, ses favoris et les tendances globales.

Inspiré des approches décrites par Gérald H. (2023), il repose sur :

- **Filtrage par contenu** : similarité entre les genres et mots-clés des films aimés.
- **Filtrage collaboratif** : comparaison des profils utilisateurs similaires.
- **Popularité globale** : prise en compte du nombre total de clics.

b) Données exploitées

- Historique de consultation.
- Films ajoutés en favoris.
- Notes et classements TMDb.

c) Algorithme de base (pseudo-code)

Pour chaque film F vu ou enregistré par l'utilisateur U :

Identifier les genres dominants

Chercher dans la base les films similaires (genre + note)

Classer les résultats par popularité

Retourner les 10 premiers films recommandés

d) Évolution possible

Intégration future d'un modèle **IA léger (TensorFlow.js)** pour prédiction du score d'affinité entre utilisateurs et films.

13. Conclusion

Le projet **My-WatchWise-VF** illustre la capacité à concevoir un système hypermédia moderne alliant technologie, ergonomie et intelligence fonctionnelle.

Malgré un délai court de trois semaines, toutes les fonctionnalités principales ont été réalisées : authentification Firebase, recherche TMDb, gestion des favoris et interface responsive.

La structure du projet, fondée sur Angular et Firebase, garantit sa pérennité, sa scalabilité et sa modularité.

Ce projet démontre la maîtrise de la chaîne complète du développement web :

- **Conception → Développement → Validation → Hébergement.**

Les perspectives d'évolution incluent :

- L'intégration d'un système de recommandation intelligent basé sur apprentissage automatique.
- L'extension du projet vers une application mobile hybride.
- L'amélioration continue de l'interface et de la personnalisation utilisateur.

Ainsi, My-WatchWise-VF s'inscrit pleinement dans la philosophie des hypermédias contemporains : interactivité, adaptativité et connectivité, tout en valorisant l'approche utilisateur au cœur du design.