Application "Jargon Informatique" –Développement Approfondi

**1. Introduction**

L’application "Jargon Informatique" a pour objectif de démystifier les termes techniques de l’informatique pour les débutants, les étudiants et les professionnels non-techniques. Elle se base sur une approche pédagogique interactive, combinant un dictionnaire enrichi, des quiz et une communauté contributive.

**Le maitre d’ouvrage et le maitre d’œuvre du projet JARGON INFORMATIQUE**

1. **Maître d'Ouvrage (MOA)**

Le maître d'ouvrage est généralement la partie qui souhaite réaliser un projet ou une application pour répondre à un besoin spécifique. Il représente les intérêts des utilisateurs finaux et est responsable de la définition des objectifs, des exigences et des spécifications fonctionnelles du projet. Autrement dit, le MOA est souvent :

L'équipe de développement est chargée de la conception, de la programmation et de la mise en place du projet. Elle doit s'assurer que le projet répond aux besoins des utilisateurs et qu'il est compatible avec les différents systèmes informatiques.

Le département informatique est responsable de la maintenance et de l'évolution du projet. Il doit s'assurer que le projet est toujours en conformité avec les normes et les réglementations en vigueur.

Les utilisateurs sont les principaux bénéficiaires du projet. Ils doivent donc être impliqués dès le début du projet afin de s'assurer que le projet répond à leurs besoins.

* **Le chef de projet Jargon Informatique**

Le chef de projet informatique est une personne qui est chargée de la gestion d'un projet informatique. Il est responsable du bon déroulement du projet et du respect des délais. Il doit également veiller à ce que le projet soit conforme aux objectifs fixés. Il doit être en mesure de gérer les différentes phases d'un projet informatique. Il doit également être en mesure de gérer les différents acteurs du projet.

Le chef de projet informatique doit avoir une bonne connaissance des outils informatiques. Il doit également avoir une bonne connaissance des différents processus de développement d'un [projet informatique](https://www.solution-forum.com/les-nouveaux-besoins-des-entreprises-face-a-l-expansion-de-l-informatique/).

* **L'équipe de développement**

L'équipe de développement est composée des programmeurs, des développeurs web, des graphistes et des experts en base de données. Ces professionnels travaillent ensemble pour concevoir, développer, tester et déployer des applications et des sites web. Ils doivent également assurer la maintenance et le support des utilisateurs.

* **Le client**

Le client est l'acteur principal d'un projet informatique. Il définit les objectifs du projet et fournit les données nécessaires à son développement. Le client peut être une entreprise, un organisme public ou un particulier. Il est représenté par le sponsor, le maître d'ouvrage ou le maître d'œuvre.

* **Le fournisseur**

Le fournisseur est l'entité qui fournit les équipements et les logiciels nécessaires au bon fonctionnement du projet informatique. Il est également responsable de la maintenance et de la mise à jour des équipements et des logiciels.

* **L'opérateur**

Dans un projet informatique, l'opérateur est l'un des principaux acteurs. Il est responsable de la mise en œuvre et de la maintenance du réseau, des systèmes et des applications. Il doit assurer la disponibilité et la fiabilité du service informatique.

* **L'utilisateur final**

L'utilisateur final est celui qui va utiliser l'application finale. Il est important de bien le considérer dans le cadre d'un projet informatique car c'est lui qui va en tirer le plus grand bénéfice. Il faut donc prendre en compte ses besoins et lui fournir une application qui répondra à ses attentes.

Le rôle de l'utilisateur final est d'être le testeur de l'application. Il va donc la tester et signaler tout problème qu'il rencontre. Il est important de bien l'écouter et de prendre en compte ses remarques car elles peuvent permettre d'améliorer l'application.

L'utilisateur final doit être impliqué dans le projet dès le début afin de mieux comprendre ses besoins. Il faut également le sensibiliser aux bonnes pratiques et lui montrer comment l'application peut lui être utile.

Enfin, il est important de garder en tête que l'utilisateur final est la personne qui va utiliser l'application au quotidien et qu'il faut donc tout faire pour lui faciliter la vie.

1. **Maître d'Œuvre (MOE)**

Le maître d'œuvre, quant à lui, est l'entité ou la personne chargée de la réalisation technique du projet. Il est responsable de la mise en œuvre des exigences définies par le MOA. Les principales caractéristiques du MOE incluent :

* **Développement technique** Il s'occupe de la conception, du développement, de la mise en œuvre et parfois de la maintenance de l'application.

**Gestion de projet** : Il organise les ressources (humaines et techniques) nécessaires à la réalisation du projet, en respectant les délais et les budgets.

* **Coordination des équipes :** Le MOE travaille souvent avec des développeurs, des designers, des testeurs, et autres parties prenantes techniques.

1. **Interaction entre MOA et MOE**

La collaboration entre le maître d'ouvrage et le maître d'œuvre est cruciale pour le succès d'un projet. Le MOA doit communiquer clairement ses besoins et attentes, tandis que le MOE doit faire preuve de transparence sur les contraintes techniques et les délais. Cette interaction garantit que le résultat final correspond aux souhaits du client tout en respectant les possibilités techniques.

En résumé, le maître d'ouvrage définit le « **quoi** » et le «**pourquoi** » d'un projet, tandis que le maître d'œuvre s'occupe du « **comment** ». Ensemble, ils forment une équipe essentielle à la réussite de tout projet informatique.

**2. Fonctionnalités Détaillées**

**2.1. Dictionnaire du Jargon Technique**

Recherche Intelligente

* Barre de recherche instantanée avec suggestions (comme Google).
* Filtres avancés :
  + Par domaine (Dev, IA, Cybersécurité, Cloud, etc.).
  + Par popularité (mots les plus recherchés).
* Prononciation audio (pour les acronymes comme "SQL", "API").

Fiches Détaillées

Chaque terme contient :  
✔ Définition simple (ex: *"Une API est un pont entre deux logiciels"*).  
✔ Exemple concret (ex: *"Quand une app météo utilise les données de Google, elle appelle son API"*).  
✔ Liens vers des ressources externes (MDN Web Docs, Stack Overflow).  
✔ Variantes (ex: "JavaScript" vs "Java").

**2.2. Système de Quiz & Apprentissage**

* Quiz personnalisés :
  + Basés sur l’historique de recherche de l’utilisateur.
  + Questions à choix multiples (QCM) ou vrai/faux.
* Mode Défi :
  + Chronométré (ex: 10 termes en 1 minute).
  + Classement global (leaderboard).
* Certificats de Maîtrise :
  + Badges numériques pour chaque catégorie maîtrisée (ex: *"Expert en Réseaux"*).

**2.3. Gestion des Favoris & Historique**

* Liste personnalisable pour sauvegarder les termes complexes.
* Synchronisation cloud (via Firebase Auth).
* Export PDF (pour révision hors ligne).

**2.4. Mode Hors Ligne**

* Base de données locale (SQLite ou Realm) pour un accès sans Internet.
* Mises à jour différées (quand la connexion revient).

**2.5. Contribution Communautaire**

* Système de vote :
  + Les utilisateurs peuvent proposer des définitions alternatives.
  + Modération par les administrateurs.
* Gamification :
  + Points de contribution (karma) pour les utilisateurs actifs.
  + Niveaux de "Expert" déblocables.

**3. Modèle de Développement (Agile/Scrum)**

**3.1. Méthodologie Agile**

* Itérations courtes (Sprints de 2 semaines) :
  + Chaque sprint livre une nouvelle fonctionnalité testée.
  + Exemple : \*Sprint 1 = Dictionnaire de base, Sprint 2 = Quiz, etc.\*
* Rôles clés :
  + Product Owner : Priorise les features (via le *Product Backlog*).
  + Scrum Master : Résout les blocages.
  + Équipe Dev : Devs Full-Stack, UX/UI Designer.

**3.2. Outils de Gestion de Projet**

* Jira ou Trello pour suivre les tâches (*To Do, In Progress, Done*).
* GitHub Projects pour le versioning et les revues de code (*Pull Requests*).
* Figma pour les maquettes UI/UX.

**3.3. Revue & Rétrospective**

* À la fin de chaque sprint :
  + Démo aux stakeholders.
  + Améliorations continues (*"Pourquoi le quiz a-t-il pris du retard ?"*).

**4. Architecture Technique**

**4.1. Frontend (Mobile & Web)**

* Framework : React Native (pour iOS, Android) + React.js (version web).
* UI/UX :
  + Composants réutilisables (ex: <TermCard />).
  + Animations fluides (Framer Motion).
* État global : Redux ou Context API.

**4.2. Backend (API & Logique Métier)**

* Langage : Node.js (Express) ou Python (FastAPI).
* Endpoints :
  + GET /api/terms?search=API → Retourne la définition.
  + POST /api/quiz/generate → Crée un quiz aléatoire.
* Authentification : JWT (JSON Web Tokens) + OAuth (Google/GitHub login).

**4.3. Base de Données**

| Besoin | Technologie | Exemple |
| --- | --- | --- |
| Données structurées | PostgreSQL | Table terms(id, name, definition) |
| Recherche full-text | Elasticsearch | Optimise la recherche ("CLI" ≈ "Command Line") |
| Données temps réel | Firebase Firestore | Synchronisation des favoris |

**4.4. DevOps & Infra**

* CI/CD :
  + GitHub Actions (automatise les tests et déploiements).
  + Docker + Kubernetes (pour la scalabilité).
* Hébergement :
  + Frontend : Vercel/Netlify.
  + Backend : AWS EC2 ou Heroku.
* Monitoring :
  + Sentry (erreurs en temps réel).
  + Google Analytics (tracking des recherches populaires).

**5. Tests & Qualité Logicielle**

* Tests Unitaires : Jest (JavaScript), Pytest (Python).
* Tests d’Intégration : Postman (vérifie que l’API renvoie bien HTTP 200).
* Tests UI : Cypress (vérifie que le bouton "Quiz" fonctionne).

**6. Conclusion**

L’application "Jargon Informatique" combine :  
🔹 Une approche Agile pour un développement flexible.  
🔹 Une stack technique moderne (React Native, Node.js, PostgreSQL).  
🔹 Des fonctionnalités centrées utilisateur (quiz, mode hors ligne, communauté).