**Cahier des charges fonctionnel – Application de vote pour les meilleurs jeux vidéo de l’année**

**1. Contexte**

Dans le secteur du jeu vidéo, de nombreuses cérémonies et sondages permettent aux joueurs d’élire les meilleurs jeux de l’année. Des événements tels que les **Steam Awards**, **The Game Awards** ou les **Golden Joystick Awards** illustrent l’engouement du public pour voter et célébrer les jeux vidéo marquants de l’année. Dans ce contexte, une organisation (par exemple un festival de jeux vidéo ou un média spécialisé) souhaite mettre en place sa propre plateforme de vote en ligne afin d’impliquer la communauté de joueurs et de désigner les titres les plus plébiscités.

Le projet consiste à développer une application web de **consultation grand public** où les joueurs peuvent, après inscription, voter pour leurs jeux favoris parmi une liste de candidats préalablement établie. Contrairement à certaines remises de prix participatives, il n’y aura pas de phase de nomination ouverte aux utilisateurs – les jeux en lice sont présélectionnés par l’organisateur. L’application s’inscrit dans une optique de **sondage massif** et ludique, visant un public large de gamers, à l’image des opérations communautaires menées sur Steam ou par des magazines de jeux vidéo. Elle doit ainsi conjuguer **simplicité d’utilisation**, **fiabilité du scrutin** et **attrait visuel**, afin d’inciter un maximum de fans à participer.

D’un point de vue professionnel, ce projet se déroule dans le cadre d’un événement annuel où l’organisateur souhaite **digitaliser entièrement le processus de vote**. Le système devra couvrir l’ensemble du cycle d’une élection en ligne : de l’inscription des votants jusqu’à la publication des résultats, en passant par la gestion des catégories et des candidats. Le développement est confié à une équipe technique dédiée, qui conçoit une solution sur mesure répondant aux besoins de l’organisation tout en respectant les contraintes du domaine (sécurité, anonymat, charge, réglementation, etc.). Le produit final sera maintenu par cette équipe interne et pourra être réutilisé lors d’éditions futures de l’événement.

**2. Contraintes**

Plusieurs contraintes fortes encadrent la réalisation de cette application, découlant tant des exigences du domaine que des impératifs techniques du projet.

**2.1 Contraintes fonctionnelles et règlementaires :** Le système de vote devra impérativement garantir un **scrutin équitable et sécurisé**. Chaque électeur n’a droit qu’à **un seul vote par catégorie**, et une fois son vote soumis il est **irréversible** . Le vote doit en outre être **strictement anonyme**, ce qui implique qu’il soit impossible de relier un bulletin à l’identité du votant . Aucune donnée personnelle d’électeur ne devra donc être associée aux choix effectués, et seules des statistiques agrégées seront exploitées en sortie. De plus, le système respecte le principe de **confidentialité totale avant clôture** : aucun résultat partiel ne doit être accessible ou affiché avant la fin du scrutin . Cela s’applique au grand public bien sûr, mais également à l’administrateur : même ce dernier ne pourra consulter les tendances ou résultats avant la clôture officielle, afin de prévenir toute fuite ou influence indésirable.

En matière de conformité, l’application devra se plier aux exigences légales telles que le **RGPD** pour la protection des données personnelles des votants. Par exemple, l’inscription nécessitera de recueillir des informations minimales (email, pseudonyme…) et d’informer l’utilisateur de l’usage de ses données (politique de confidentialité). Les données personnelles seront stockées de façon sécurisée et ne seront utilisées qu’aux fins du vote (authentification, suivi de participation), avec suppression ou anonymisation une fois la consultation terminée si nécessaire. Si le public visé inclut des mineurs, des mesures supplémentaires (consentement parental, conformité **RGPD-K**) devront être envisagées .

**2.2 Contraintes techniques :** Le choix technologique se portera sur une **architecture web client-serveur** classique. L’application sera développée en environnement **full web (HTML/CSS/JS côté client et PHP/MySQL côté serveur)**, conformément aux standards de la plateforme visée . Une base de données relationnelle robuste est requise pour stocker les comptes utilisateurs, les jeux candidats et les votes enregistrés. Le code devra être maintenable et versionné (utilisation de Git pour le suivi des versions et la collaboration entre développeurs). Par ailleurs, une attention particulière est portée à la **sécurité applicative** : gestion des sessions et des permissions, chiffrement des mots de passe (hashage), connexion HTTPS, protection contre les injections SQL et les failles XSS, etc. . Aucune ressource sensible (pages d’administration, scripts de traitement) ne devra être accessible sans contrôle d’accès. L’interface d’administration elle-même sera sécurisée (authentification renforcée, éventuellement filtrage par rôle ou par adresse IP) pour éviter toute intrusion .

**2.3 Contraintes d’utilisation et d’ergonomie :** Étant donné le public large visé, l’interface utilisateur doit être **claire, intuitive et ergonomique**. L’inscription et le vote doivent pouvoir être effectués facilement, y compris par des personnes peu familières avec des applications complexes. On veillera à soigner l’**expérience utilisateur** : pages épurées, instructions visibles, navigation logique entre les étapes (connexion → vote → confirmation). L’interface sera **responsive** pour être utilisable sur mobile, tablette ou ordinateur, sans développement d’application native dédiée (l’accès se fait via le navigateur). De plus, le design suivra les bonnes pratiques d’**accessibilité** (contrastes de couleurs adaptés, textes alternatifs, navigation clavier, etc.) afin de ne pas exclure les personnes en situation de handicap. L’accessibilité et la compatibilité multi-plateformes sont d’autant plus cruciales que le vote est public et doit toucher un maximum de participants.

**2.4 Contraintes de calendrier et de ressources :** Le projet s’inscrit dans un calendrier déterminé (voir section 6) qui impose une organisation rigoureuse du travail. L’équipe de développement est de taille réduite (deux personnes) et doit respecter des échéances pédagogiques précises. Ceci limite le périmètre fonctionnel réalisable en quelques mois et nécessite de faire des choix (prioriser les fonctionnalités essentielles du vote en ligne, reporter les idées annexes). Enfin, l’application devant potentiellement accueillir un grand nombre de votants en un laps de temps court (pics de connexion lors de l’ouverture ou juste avant la clôture du vote), il faudra prendre en compte des **contraintes de charge**. Cela pourra influencer des décisions techniques (optimisation des requêtes SQL, hébergement sur un serveur dimensionné en conséquence, tests de montée en charge) pour assurer la **fiabilité** du service lors des pointes de trafic.

**3. Description fonctionnelle des besoins**

**A. Cadre du projet**

Ce projet de plateforme de vote vidéoludique se déroule dans un **cadre professionnel simulé** : on peut l’assimiler à l’initiative d’un organisateur d’événement ou d’un média du secteur du jeu vidéo souhaitant impliquer la communauté. Par exemple, il pourrait s’agir d’un festival de jeux qui veut permettre au public d’élire les titres marquants de l’année, ou d’un site web spécialisé lançant un sondage en ligne pour décerner des « Awards » communautaires. Le développement est pris en charge par une petite **équipe technique** dédiée (interne ou prestataire) qui a pour mission de concevoir, réaliser et déployer l’application dans les délais impartis. Cette équipe travaillera en étroite collaboration avec l’organisateur (le **client**) afin de bien comprendre le besoin métier et d’adapter l’outil au contexte (catégories de prix spécifiques, identité visuelle de l’événement, etc.).

En se plaçant dans ce contexte, l’objectif global du projet est de **dématérialiser le processus de vote** pour un concours de jeux vidéo. L’application remplacera avantageusement un vote papier ou un sondage en ligne rudimentaire, en apportant plus de **sécurité**, de **contrôle** et de **richesse fonctionnelle**. Elle doit permettre à l’organisateur de gérer facilement l’élection (ajout des jeux en compétition, suivi de la participation, calcul automatique des résultats) et aux joueurs de voter de n’importe où via internet. Le cadre du projet implique également un enjeu de **visibilité** : une telle plateforme, si elle est réussie, peut renforcer l’engagement de la communauté de joueurs et donner de l’ampleur médiatique à l’événement (par le partage des votes sur les réseaux sociaux, l’anticipation des résultats, etc.). Ainsi, au-delà d’un simple outil interne, le produit fini s’inscrit dans la stratégie de communication de l’organisateur.

**B. Expression du besoin**

Le besoin principal exprimé par l’organisateur est de **permettre à chaque joueur de voter en ligne** pour ses jeux favoris de l’année, de manière simple, fiable et sécurisée. Actuellement, sans cette plateforme, le vote du public serait soit inexistant, soit géré de façon peu pratique (formulaires manuels, sondages dispersés). L’objectif est donc de disposer d’une **application web centralisée** qui gère l’élection de bout en bout. Les **résultats attendus** du projet sont : (1) une participation massive et fluide des joueurs au vote, (2) un dépouillement automatisé produisant le palmarès des jeux gagnants, et (3) une expérience utilisateur positive renforçant la confiance des votants dans le processus.

Plusieurs **fonctionnalités clés** découlent de ce besoin : tout d’abord, une **authentification sécurisée** des électeurs, afin de s’assurer que seuls les votants autorisés participent et qu’un même joueur ne puisse pas voter plusieurs fois. Ensuite, une **interface de vote** intuitive présentant les catégories de récompense (exemple : *Meilleur jeu d’action*, *Meilleur jeu indépendant*, *Jeu de l’année*, etc.) et les nominés dans chaque catégorie, pour que l’utilisateur puisse sélectionner son choix facilement. Le système doit **enregistrer chaque vote** de façon fiable, sans possibilité de modification une fois soumis (afin de préserver l’intégrité du scrutin). Il doit également **empêcher le vote en dehors de la période autorisée** : avant l’ouverture, l’utilisateur ne doit pas accéder au bulletin, et après la clôture le vote doit être désactivé automatiquement . Pendant la phase active, l’application peut indiquer à l’électeur si son vote a bien été pris en compte et éventuellement lui permettre de consulter un récapitulatif de sa participation (par exemple, les catégories pour lesquelles il a voté, sans révéler évidemment son choix précis, pour rester neutre).

Du point de vue de l’organisateur, le besoin inclut une **console d’administration** pour piloter le scrutin. L’administrateur doit pouvoir configurer les paramètres du vote : créer ou éditer les catégories, **enregistrer les candidats** (jeux en compétition) pour chaque catégorie, gérer la **liste des électeurs** (par exemple, modérer les inscriptions, éventuellement ajouter manuellement des comptes vérifiés), et définir les **dates d’ouverture et de clôture** du vote . Ce panneau d’administration permettra également de **déclencher la clôture** le moment venu et de **publier les résultats**. Les résultats, sous forme de **données agrégées**, devraient s’afficher de manière compréhensible (tableaux de scores par catégorie, pourcentages, graphiques éventuels) pour communiquer les gagnants et les statistiques globales au public . L’organisateur a exprimé le souhait de *valoriser ces résultats* lors d’une cérémonie ou d’une annonce publique, il est donc crucial que l’application offre un **module de consultation des résultats** attractif et accessible à tous une fois le vote terminé.

En résumé, les **objectifs du projet** sont :

* **Donner la possibilité aux joueurs de voter** en ligne pour les meilleurs jeux de l’année (une voix par personne et par catégorie).
* **Garantir un vote éthique et fiable**, respectant l’anonymat des votants et empêchant toute triche (multivotes, usurpation d’identité, etc.).
* **Faciliter le travail de l’organisateur** grâce à des outils d’administration (gestion des utilisateurs, des jeux, et supervision du déroulement du scrutin).
* **Automatiser le dépouillement** et la publication des résultats afin d’éviter les erreurs humaines et d’obtenir rapidement le palmarès une fois le vote clos.
* **Offrir une expérience utilisateur** agréable tant pour les votants (simplicité, rapidité, aspect ludique du vote) que pour l’administrateur (interface de gestion claire, feedback sur l’état du scrutin).
* **Assurer la conformité** du système vis-à-vis des bonnes pratiques de développement (code de qualité, documentation) et des normes en vigueur (sécurité informatique, protection des données, accessibilité numérique).

**C. Analyse de l’existant**

Plusieurs solutions ou événements existants ont été étudiés afin de dégager les bonnes pratiques pour cette plateforme de vote :

* **The Game Awards** : Il s’agit d’une cérémonie annuelle américaine très médiatisée. Côté public, les fans peuvent voter via le site officiel en se connectant avec un compte social (X/Twitter, Facebook, Twitch ou Google) . Notons que dans ce cas, le vote du public n’influence qu’en partie le résultat (à hauteur de 10% des votes, le reste étant décidé par un jury professionnel) . Néanmoins, ce modèle montre l’importance d’une **authentification simplifiée** (via des comptes existants) pour toucher un large public, et prouve que l’on peut combiner un **vote en ligne grand public** avec des mécanismes de modération (ici un jury). Pour notre projet, nous retiendrons surtout l’idée de proposer une connexion rapide pour les votants (au minimum par email, voire via des OAuth tiers si envisagé ultérieurement) et l’idée qu’une part du vote peut être réservée à un **rôle administrateur/jury** si jamais l’organisateur voulait garder un contrôle (dans notre cas, ce n’est pas prévu car le vote est 100% public).
* **Steam Awards** : Cette initiative de Valve sur la plateforme Steam est un **vote communautaire massif** en deux phases. Les utilisateurs Steam nominent d’abord librement des jeux dans des catégories définies par Valve, puis les organisateurs sélectionnent les finalistes et soumettent un vote final à tous les joueurs  . Ce processus démontre la capacité à gérer une participation énorme (des millions de votes) sur une infrastructure web robuste. Il montre aussi l’intérêt de **catégories originales** pour impliquer la communauté. Cependant, le Steam Awards n’insiste pas particulièrement sur l’anonymat du vote – Valve enregistre les votes des utilisateurs sur son système interne – ce qui nous conforte dans l’idée d’apporter un **surcroît de confidentialité** dans notre propre solution. En outre, comme nous n’aurons pas de phase de nomination libre, notre système sera plus simple (une seule phase de vote final), tout en gardant l’esprit communautaire (un compte Steam étant l’équivalent chez nous d’un compte utilisateur sur la plateforme).
* **Golden Joystick Awards** : L’un des plus anciens prix jeux vidéo britanniques, entièrement décidé par les joueurs. Chaque année, une vingtaine de catégories sont proposées (meilleur jeu, meilleure narration, etc.) et le public soumet des millions de votes via le site web officiel . Les organisateurs des Golden Joystick offrent même des **incentives** pour encourager le vote, par exemple en 2025 chaque votant pouvait réclamer un ebook gratuit après avoir voté . Cette plateforme confirme le modèle du **vote 100% public à grande échelle**. Elle met en avant la nécessité d’avoir une **infrastructure capable de gérer un grand nombre de catégories et de votes**, ainsi qu’une **interface attrayante** (sur le site Golden Joystick, le vote se fait via une page web dédiée avec la liste des catégories, et il faut enregistrer ses votes via un formulaire avec email pour validation  ). De cet exemple, nous retenons qu’il est pertinent de proposer **plusieurs catégories de prix** dans le même système et de prévoir une **confirmation de vote par l’utilisateur** (par exemple un bouton de validation finale, voire un email de confirmation comme l’exige Golden Joystick pour attribuer une récompense). Notre projet s’inspire de ces bonnes pratiques : multiplicité des catégories, processus de vote simple et rapide, et éventuellement offre d’un feedback positif au votant (message de remerciement, goodies numériques, etc., si le contexte le permet).

En synthèse, **l’analyse concurrentielle** des solutions existantes guide plusieurs choix pour notre application. D’une part, elle souligne l’importance d’un **système de comptes utilisateurs** pour contrôler l’accès au vote (tous les exemples utilisent une forme de connexion compte/joueur). D’autre part, elle confirme que le **vote par catégories multiples** est un standard attendu pour couvrir les “meilleurs jeux” sous différents angles – notre application intègrera donc la notion de catégories configurables pour les récompenses . Nous constatons également qu’aucune des plateformes étudiées ne met l’accent sur l’anonymat absolu du vote : c’est un **point différenciant** que nous souhaitons apporter, en stockant les bulletins de manière dissociée de l’identité des votants. Enfin, tous ces systèmes fonctionnent sur un principe de **calendrier défini** (ouverture et fermeture du vote annoncées à l’avance, résultats publiés lors d’un événement fixé) – cela nous conforte dans la nécessité d’implémenter une gestion temporelle rigoureuse et d’empêcher toute consultation prématurée des résultats . L’étude de l’existant n’a pas révélé de solution clé-en-main correspondant exactement à notre besoin, ce qui justifie pleinement le développement spécifique de cette application en combinant les meilleures pratiques observées.

**D. Acteurs du projet**

Plusieurs acteurs interviennent dans le cadre de ce projet, chacun jouant un rôle et des attentes spécifiques :

* **Électeurs** : Ce sont les joueurs, membres du public, qui vont utiliser la plateforme pour voter. Leur rôle se limite à créer un compte sur le site, à **s’authentifier** et à **participer au vote** pendant la période d’ouverture. Ils interagissent donc avec l’interface grand public de l’application (pages d’inscription, de connexion, puis bulletin de vote en ligne). Leurs attentes principales sont la simplicité (processus d’inscription rapide, vote en quelques clics) et la transparence (confirmation que leur vote a bien été pris en compte, garanties sur la confidentialité). Du point de vue des exigences, les données personnelles fournies par les électeurs (nom, prénom, email, etc.) ne seront utilisées qu’à des fins d’identification/authentification et **doivent rester strictement confidentielles vis-à-vis du contenu du vote** (anonymat).
* **Administrateur / Organisateur** : Il s’agit de la personne (ou de l’équipe) responsable de la gestion du scrutin en ligne. Cet acteur dispose d’un **espace administrateur sécurisé** pour configurer l’élection. Ses tâches incluent : définir les catégories de prix, **enregistrer les jeux candidats** pour chaque catégorie, gérer les comptes des votants (valider ou supprimer des inscriptions si nécessaire), et paramétrer les dates d’ouverture/clôture du vote. L’administrateur a aussi la charge de déclencher la **publication des résultats** une fois le vote terminé. Il a donc des droits élevés sur l’application, avec accès à des fonctionnalités cachées au public. Ses attentes sont d’avoir un **outil d’administration ergonomique et complet**, lui évitant des manipulations techniques lourdes (il ne doit pas avoir à intervenir dans la base de données manuellement, tout doit se faire via l’UI admin). Il souhaite par ailleurs que l’application fournisse des **indicateurs de suivi** durant le déroulement du vote (par exemple, nombre d’inscrits, taux de participation global en temps réel) sans compromettre l’anonymat ni révéler les tendances. Enfin, en tant qu’organisateur de l’événement, cet acteur est garant du bon déroulement et de la crédibilité du vote, il attend donc du système une **sécurité sans faille** (pas de risque de vote multiple ou de fraude informatique).
* **Développeurs (équipe technique)** : Ce sont les personnes chargées de concevoir, développer et maintenir l’application de vote. Dans un contexte réel, il pourrait s’agir d’un service informatique interne ou d’un prestataire externe. Dans le cadre de ce projet, l’équipe de développement est composée de deux étudiants en informatique travaillant en binôme. Leurs responsabilités englobent la **spécification** (rédaction de ce cahier des charges), la **conception** (modélisation de la base de données, architecture logicielle, interfaces), le **développement** du front-end et du back-end, ainsi que les **tests** et la mise en production sur un serveur web. Ils doivent aussi veiller à la **qualité du code** (commentaires, documentation, suivis de versions) et au respect des bonnes pratiques de sécurité et d’ergonomie. Bien qu’ils n’interagissent pas en tant qu’utilisateurs finaux de l’application, ils sont des acteurs clés du projet en ce qu’ils vont traduire les besoins métier en une solution technique opérationnelle. Leur succès sera mesuré à la satisfaction des autres parties prenantes (organisateur et électeurs) ainsi qu’au respect des délais et critères de qualité définis initialement.
* **Autres parties prenantes** : On peut citer enfin l’**organisme commanditaire** (par ex. le comité du festival ou la rédaction du média qui organise le vote) et les **référents légaux/qualité** éventuellement impliqués. Le commanditaire est celui qui définit les objectifs initiaux du projet et finance/décide de son lancement – son intérêt est de récupérer un outil fonctionnel répondant à ses besoins et d’obtenir des **résultats exploitables** (les vainqueurs du vote) pour communication. Il ne va pas utiliser directement l’application hormis pour consulter les résultats finaux publiés, mais il valide les grandes orientations (choix des catégories, liste des nominés, etc.). Les conseillers juridiques, quant à eux, peuvent intervenir pour vérifier la conformité légale (rédaction des conditions d’utilisation, respect des règles du jeu concours s’il y a lieu, protection des données…). Enfin, on peut inclure comme parties prenantes indirectes les **éditeurs de jeux vidéo candidats** : même s’ils n’ont pas de rôle actif dans le système, ils sont concernés par l’issue du vote. L’organisateur devra donc veiller à ce que la plateforme inspire confiance et sérieux, car l’image de l’événement et de ses partenaires (éditeurs, sponsors) en dépend.

**4. Périmètre du projet**

Le périmètre définit les fonctionnalités qui seront prises en charge par l’application et celles qui en sont exclues, afin de cadrer les travaux et les attentes.

**Fonctionnalités incluses (ce qui sera fait) :**

* **Inscription et authentification sécurisées des utilisateurs :** les joueurs pourront créer un compte (inscription avec email, mot de passe, pseudo…), confirmer éventuellement leur email, puis se connecter à l’aide de leurs identifiants. La gestion de session sécurisée (cookies de session, timeout d’inactivité) sera implémentée pour maintenir la connexion de l’électeur dans son espace privé.
* **Gestion des catégories et des candidats :** l’administrateur disposera d’une interface pour créer, modifier ou supprimer les **catégories de vote** (par exemple ajouter une catégorie “Meilleur jeu de stratégie”) et pour ajouter/éditer la liste des **jeux nominés** dans chaque catégorie. Ces données seront stockées en base et dynamiseront l’interface de vote.
* **Interface de vote en ligne :** côté électeur, une page dédiée permettra de voter. L’utilisateur verra la liste des catégories ouvertes au vote, avec pour chacune les jeux en lice, et pourra sélectionner son choix (par ex. via un bouton radio ou une carte à cliquer). L’UI sera conçue pour présenter une catégorie à la fois ou toutes à la suite, de manière claire. Une fois tous ses choix faits, il validera son bulletin. Un accusé (message de confirmation) sera affiché pour récapituler sa participation.
* **Vote unique et stockage anonyme des bulletins :** le système fera respecter la règle “un électeur = un vote par catégorie”. Cela implique qu’à la validation, une vérification empêche l’utilisateur de soumettre une catégorie deux fois ou de revenir modifier son vote. Chaque bulletin sera enregistré de façon **anonymisée** : la base de données stockera les votes sans indiquer l’utilisateur associé, ou en séparant dans des tables différentes les liens entre comptes et votes, de sorte qu’aucune requête simple ne puisse lier un électeur à son choix.
* **Gestion des phases du scrutin :** le logiciel intégrera la notion de **phase temporelle**. Avant l’ouverture officielle, toute tentative d’accès à la page de vote sera redirigée vers un message “Le vote n’est pas encore ouvert, rendez-vous à telle date/heure”. À l’heure d’ouverture, le vote devient accessible automatiquement (sans intervention manuelle, grâce à un champ date ou une tâche planifiée côté serveur). À la **clôture**, les formulaires de vote seront verrouillés (plus possible de voter) et l’application passera en mode consultation des résultats. On veillera à ce qu’après clôture, **aucun nouveau vote ne puisse être pris en compte** même si un utilisateur malintentionné tente d’appeler une API ou de bricoler le client.
* **Calcul et affichage des résultats :** une fois le vote clos, l’application compilera les résultats. Elle calculera le nombre total de voix par jeu et par catégorie, déterminera le gagnant de chaque catégorie (le jeu ayant le plus de votes, ou ex-æquo le cas échéant), et produira des **statistiques globales** (par exemple le taux de participation, le nombre total de votants, etc.). Ces résultats seront présentés sur une page publique de l’application, accessible à tous. L’affichage sera soigné : liste des catégories avec pour chacune le jeu gagnant mis en avant, éventuellement un podium (top 3) ou un pourcentage des voix pour information. Des **graphiques** ou diagrammes pourront illustrer la répartition des votes par catégorie si cela apporte un plus de lisibilité (option envisagée si le temps le permet).
* **Publication contrôlée des résultats et gestion des accès :** l’administrateur aura la possibilité, via sa console, de vérifier les résultats calculés puis de lancer leur publication officielle. Une fois publiés, ceux-ci deviennent visibles par tous les utilisateurs du site (même non connectés). Avant cela, les résultats restent inaccessibles (écran “en attente” ou compte à rebours vers l’annonce). Cette fonctionnalité garantit que l’organisateur maîtrise le moment de l’annonce des vainqueurs (par exemple, aligné avec une cérémonie ou un communiqué de presse).
* **Interfaces ergonomiques pour tous les rôles :** le projet comprend la conception d’une **interface web** adaptée aux électeurs (site vitrine avec accueil, pages de vote, contact) et à l’administrateur (interface de back-office). L’accent sera mis sur la cohérence graphique, la simplicité de navigation et l’ergonomie générale pour satisfaire chaque utilisateur du système, qu’il soit votant ou gestionnaire.

**Fonctionnalités exclues (ce qui ne sera pas fait) :**

* *Modules communautaires avancés* : il n’est pas prévu d’intégrer un système de **commentaires**, forum ou chat au sein de la plateforme de vote. Les interactions entre utilisateurs se limiteront éventuellement à un bouton de partage sur les réseaux sociaux, mais aucune discussion publique n’a lieu sur le site de vote lui-même.
* *Modes de vote complexes* : le scrutin sera volontairement **simple (majoritaire à un tour)**. Ainsi, aucune mécanique de vote alternatif ne sera implémentée : pas de vote pondéré par note, pas de système de **préférences multiples** ou de classement de jeux, pas de second tour. Chaque catégorie aura un vainqueur déterminé uniquement par le nombre de voix obtenues au terme du vote, point final.
* *Application mobile native* : le projet se concentre sur l’application web responsive accessible via navigateur. Aucune application mobile iOS/Android séparée ne sera développée dans ce cadre, afin de concentrer l’effort sur le cœur du système. Le site web sera cependant optimisé pour mobile, rendant l’app native moins nécessaire.
* *Statistiques approfondies ou temps réel* : hormis l’affichage des résultats finaux (et possiblement quelques graphiques de répartition), on n’implémentera pas de tableau de bord analytique sophistiqué. Par exemple, pas de suivi en temps réel de la progression des votes, pas d’historiques ou tendances avant/après clôture, ni d’outils de data visualisation complexes au-delà du besoin de base. L’objectif principal est de désigner les gagnants, pas de fournir une analyse détaillée des votes (bien que cela puisse être une amélioration future).
* *Fonctionnalités hors-sujet* : tout ce qui sort du domaine “vote en ligne” est écarté. Il n’y aura pas de gestion de paiements (puisque le vote est gratuit), pas de module d’envoi d’**emails de campagne** aux inscrits (en dehors d’éventuels mails techniques comme la confirmation d’inscription), pas de prise en charge multilingue avancée du site (une seule langue de référence, le français, est prévue pour ce projet éducatif), ni d’autres fonctions éloignées de l’objectif (par exemple, pas de nomination de candidats par les utilisateurs, pas de récompense matérielle automatisée aux votants hormis un message de remerciement). En somme, le périmètre est circonscrit strictement aux besoins du vote afin de garantir la faisabilité dans le temps imparti.

**5. Organisation du projet**

Pour mener à bien ce projet dans les délais fixés, une **organisation rigoureuse** a été mise en place, articulée autour de l’équipe de développement et du commanditaire.

**Équipe projet et rôles :** L’équipe de réalisation est composée de deux développeurs (étudiants en BUT Informatique 2e année) qui se répartissent les tâches. Bien que le travail soit collaboratif, chacun peut se voir attribuer des responsabilités spécifiques en fonction des compétences : par exemple, un développeur référent pour le **front-end** (conception de l’interface web, intégration HTML/CSS, ergonomie) et l’autre référent pour le **back-end** (logique métier, base de données, sécurité). Un **chef de projet** (rôle assumé collectivement ou par l’un des deux membres) est chargé de la coordination globale : suivi de l’avancement, respect du cahier des charges et communication avec le référent pédagogique (ou le client fictif). Par ailleurs, des **intervenants extérieurs** peuvent être sollicités ponctuellement : l’enseignant encadrant joue le rôle de **client** et de conseiller, validant les livrables à chaque étape clé, tandis qu’un tuteur technique peut assurer le rôle de **superviseur qualité** (revue du code, conseils d’architecture, etc.). Cette répartition s’apparente à une matrice **RACI** où les développeurs sont Responsables de la réalisation, le tuteur est Consulté pour avis technique, l’enseignant/client est Approuvé/décideur sur les grandes orientations, etc.

**Méthodologie de travail :** Le développement suit une approche **itérative** inspirée des méthodes agiles. Au lieu de tout développer d’un bloc, le projet est découpé en **plusieurs phases** avec des objectifs intermédiaires (voir la section 6 pour le planning détaillé des jalons). Chaque phase donne lieu à un livrable ou une démonstration partielle : par exemple, un prototype minimal (MVP 1) avec les fonctionnalités de base d’authentification et de vote, puis une version améliorée (MVP 2) intégrant l’admin et les résultats. Des réunions d’avancement (ou points hebdomadaires) sont prévues pour faire le bilan, identifier les blocages et ajuster le plan si nécessaire. L’équipe utilise des outils de gestion de projet pour organiser ses tâches, par exemple un tableau **Trello** ou GitHub Projects listant le backlog des user stories et les tâches à réaliser (to-do list). Une **WBS** (Work Breakdown Structure) a été esquissée en début de projet pour identifier toutes les tâches à mener (conception BD, développement de chaque fonctionnalité, tests, documentation, etc.), puis celles-ci ont été planifiées sur le calendrier.

**Outils et qualité :** Le travail collaboratif s’appuie sur un dépôt **Git** commun, hébergé sur GitHub, pour le versioning du code. Chaque développeur y pousse régulièrement ses commits en veillant à écrire des messages clairs, ce qui permet de tracer l’évolution du projet et d’assurer une intégration continue. Des conventions de code communes ont été définies (indentation, nomenclature des variables, structure MVC, etc.) afin de garantir l’homogénéité et la lisibilité du logiciel. En parallèle, l’équipe documente son travail : rédaction de **commentaires** dans le code, maintien d’un **journal de bord** pour noter les décisions importantes, et production en fin de projet d’un **rapport technique** décrivant l’architecture et les tests effectués. La qualité du développement est un aspect pris au sérieux, conformément aux attendus (tests fonctionnels, validation des exigences de sécurité, inspection par le tuteur qualité). Enfin, l’organisation inclut la préparation de la **soutenance finale** : réalisation d’une présentation (diaporama et/ou démonstration vidéo de l’application) et d’un support visuel (par exemple une affiche promotionnelle du projet) pour valoriser le travail accompli. Ces éléments de communication seront prêts en temps utile pour appuyer la livraison du projet.

**6. Délais de réalisation**

Le projet suit un calendrier précis, étalé sur environ quatre mois, avec des **jalons** (milestones) à atteindre à des dates fixées. Ci-dessous, les principales étapes et leurs échéances :

* **23 octobre 2025 – Cahier des charges fonctionnel :** Date de livraison de ce document. Il s’agit du premier livrable majeur, validant la compréhension du besoin et la définition des spécifications du projet. Après remise, une relecture avec le client/encadrant permettra d’apporter d’éventuelles corrections avant de passer à la phase de conception.
* **26 octobre 2025 – Design préliminaire (UML & MCD) :** À la fin du mois d’octobre, l’équipe doit fournir les premiers diagrammes de conception : schéma de **base de données** (MCD/MLD) pour organiser les tables utilisateurs, votes, jeux, etc., et éventuellement des diagrammes UML (cas d’utilisation, diagramme de classes principal). Ce **dossier d’analyse et de conception** validera l’architecture envisagée avant de coder  .
* **22 novembre 2025 – Prototype MVP 1 :** Environ un mois plus tard, une première version **Minimale Viable** de l’application doit être opérationnelle . Ce MVP 1 inclura typiquement les fonctionnalités essentielles : inscription/connexion des utilisateurs, enregistrement d’un vote simple, et peut-être un début d’interface admin rudimentaire. L’objectif est de démontrer un **flux de vote bout-en-bout** fonctionnel en conditions réelles (sur un serveur de test), même si l’interface est encore basique et toutes les fonctions ne sont pas encore implémentées. Ce jalon permet d’obtenir un feedback précoce et d’ajuster les priorités pour la suite.
* **12 décembre 2025 – Prototype MVP 2 :** À cette date, une seconde itération du prototype est attendue . Le MVP 2 devrait intégrer la plupart des fonctionnalités finales : interface utilisateur améliorée, espace administrateur complet (gestion catégories/jeux, contrôle du scrutin), anonymisation effective des votes, affichage des résultats post-scrutin, etc. L’ergonomie et la sécurité auront également été renforcées. En somme, c’est une version quasi-complète, proche du produit final, qui pourra servir de base aux tests utilisateurs et à la correction des derniers bugs.
* **18 janvier 2026 – Livrables finaux :** Mi-janvier, l’ensemble du travail doit être achevé. Plusieurs livrables documentaires sont prévus ce jour-là  , notamment un **rapport technique de sécurisation** (décrivant les mesures de sécurité implémentées, les tests d’intrusion éventuellement menés), le **rapport final de réalisation** (documentation utilisateur, manuel admin, bilan des fonctionnalités implémentées) et un **rapport sur la qualité** (analyse de la conformité du code, des tests, de la gestion de projet vis-à-vis des critères qualité initialement posés). C’est également la date butoir pour peaufiner l’application elle-même : correction ultime de bugs, optimisation du code, et gel de la version qui sera présentée.
* **23 janvier 2026 – Soutenance et livraison définitive :** Cette date marque la **présentation finale** du projet devant le jury (encadrants et éventuellement partenaires) . L’équipe y fera la démonstration de l’application de vote fonctionnelle, exposera les résultats obtenus (ex: nombre de votants simulés, retours d’expérience) et reviendra sur les aspects marquants de la gestion du projet. La soutenance s’accompagne de la remise des livrables définitifs (code source sur le dépôt Git, documentation finale). C’est à cette occasion que le projet est officiellement **livré au commanditaire**, prêt à être utilisé lors de l’événement réel si tel était le but.

Ce **macro-planning** est également résumé dans un diagramme de Gantt (fourni en annexe du dossier) pour visualiser le chevauchement des phases et la répartition du travail dans le temps. Chaque étape clé s’inscrit dans la progression globale du projet, depuis l’analyse des besoins jusqu’à la livraison. Le respect de ces délais est crucial pour tenir la cadence du projet (pédagogique) et s’assurer que l’application soit prête à temps pour l’échéance de l’événement annuel. Des points de contrôle réguliers avec l’encadrement permettront de valider l’atteinte de chaque jalon et d’anticiper d’éventuels ajustements de planning si nécessaire. En fixant ainsi une date de livraison attendue et des jalons intermédiaires, on balise clairement le projet et on se dote d’indicateurs de progression, gages de sérieux dans la conduite de ce développement logiciel.