

# Thankward Project Documentation

## **Team:**

Prajwal JHA

Yassine TAMIM

Abhinay REDDY

Imane BICHRI

Abdoulaye DOUMBIA

## Index

1. <a href="#">Index</a>	2
2. <a href="#">Product Backlog</a> [EN]	3
3. <a href="#">Incrément de Produit</a> [EN]	4
4. <a href="#">Definition of Done</a> [EN]	10
5. <a href="#">Documentation technique</a> [FR]	12
6. <a href="#">Manuels utilisateurs et guide de prise en main</a> [FR]	19
7. <a href="#">Rapports de Sprint et Revue de Sprint</a> [EN]	27
8. <a href="#">Retrospectives de Sprint</a> [EN]	32
9. <a href="#">Rapport de performance</a> [EN]	38
10. <a href="#">Tests et Validation</a> [FR]	43
11. <a href="#">Données sur la maintenance et support</a> [EN]	49

## Getting started

Lien dépôt Back: ( <https://github.com/ennajahayou/tkwbackendCDL.git> )

Lien dépôt Front : ( <https://github.com/ennajahayou/tkwfrontendCDL.git> )

Lien déploiement Back : ( <https://tkwbackendcdl.onrender.com> )

Lien déploiement Front : ( <https://gleaming-cactus-1facf8.netlify.app/> )

Identifiant : eman@gmail.com

password : 12345678

## Product Backlog

Link: [+ Backlog CLDL \(Sheet 2\)](#)

3

## **Incrément de Produit**

Period: September 24 - November 11, 2024

### **Database Foundation (Sprint 1)**

#### **Core Database Structure**

##### **Status: In Production**

- Common project database implemented
  - MySQL database with normalized schema
  - Tables: Users, Ideas, Feedback, Thanks, Projects
  - Foreign key constraints and indexes optimized
- Validation Status:
  - ✓ Schema validation complete
  - ✓ Data integrity tests passed
  - ✓ Performance benchmarks met

### **Idea Management System - Phase 1**

##### **Status: In Production**

- Idea Submission Form
  - React-based dynamic form
  - Field validation implemented
  - File attachment capability
  - Validation Status:
    - ✓ Form validation tests complete
    - ✓ Cross-browser compatibility verified
    - ✓ Accessibility standards met
- Draft System
  - Draft restoration
  - Version tracking
  - Validation Status:
    - ✓ Draft persistence verified

- ✓ Concurrent editing handled
- ✓ Recovery scenarios tested

## Feedback System (Sprint 2)

### Voting Mechanism

#### Status: In Production

- Binary voting system (Yes/No)
  - Real-time vote counting
  - User vote tracking
  - Vote modification handling
  - Validation Status:
    - ✓ Vote integrity tests passed
    - ✓ Race condition handling verified
    - ✓ Performance under load tested

### Feedback Processing

#### Status: In Production

- Feedback Form
  - Structured feedback collection
  - Validation Status:
    - ✓ Form submission tests complete
    - ✓ Data sanitization verified
    - ✓ Performance metrics met

### Time Management

#### Status: In Production

- 72-hour Feedback Window
  - Automated timing system
  - Timezone handling
  - Grace period management

## Thankward Project Documentation

- Validation Status:
  - ✓ Timer accuracy verified
  - ✓ Edge case handling tested
  - ✓ Notification system integrated

### **Gamification System (Sprint 3)**

#### **Thanks System**

##### **Status: In Production**

- Points Distribution
  - Automated scoring algorithm
  - Real-time updates
  - Historical tracking
  - Validation Status:
    - ✓ Point calculation accuracy verified
    - ✓ Transaction handling tested
    - ✓ Performance metrics met

#### **Project Management - Phase 1**

##### **Status: In Production**

- Project Creation
  - Template-based creation
  - Custom field support
  - Validation Status:
    - ✓ Project creation flow tested
    - ✓ Template system verified
    - ✓ Performance benchmarks met

### **Integration Phase (Sprint 4)**

#### **Organization Management**

##### **Status: In Production**

## Thankward Project Documentation

- Project Organization
  - Hierarchical structure
  - Permission management
  - Resource allocation
  - Validation Status:
    - ✓ Organization hierarchy tested
    - ✓ Permission inheritance verified
    - ✓ Resource constraints validated

### Frontend Integration - Phase 1

#### Status: In Production

- Core Pages
  - Ideas listing
  - Feedback interface
  - Project dashboard
  - Validation Status:
    - ✓ UI/UX testing complete
    - ✓ Responsive design verified
    - ✓ Performance metrics met

### User Management (Sprint 5)

#### Authentication System

#### Status: In Production

- User Authentication
  - Cookie-based auth
  - Role-based access control (CEO/Talent)
  - Session management
  - Validation Status:
    - ✓ Security testing complete
    - ✓ Performance testing passed

## Profile Management

### Status: In Production

- User Profiles
  - Profile customization
  - Activity tracking
  - Preference management
  - Validation Status:
    - ✓ Profile CRUD operations tested
    - ✓ Data consistency verified
    - ✓ Privacy controls validated

## System Integration (Sprint 6)

### UI Implementation

### Status: In Production

- Figma Design Implementation
  - Component library
  - Theme system
  - Animation framework
  - Validation Status:
    - ✓ Visual regression testing complete
    - ✓ Performance benchmarks met

## Production Deployment (Sprint 7)

### Deployment

### Status: In Production

- Production Environment
  - Container orchestration
  - CI/CD pipeline
  - Monitoring setup



## Thankward Project Documentation

- Validation Status:
  - ✓ Deployment pipeline verified
  - ✓ Production environment tested
  - ✓ Monitoring systems active

### Ongoing Development

#### Status: In Progress

- Code Optimization
  - Refactoring
  - Documentation
  - Performance optimization
  - Current Status:
    - ⇄ Code review in progress
    - ⇄ Documentation updates ongoing
    - ⇄ Performance tuning continuing
    - ✓ Deployment issues rectified

### Critical Features in Development

1. CEO Approval Workflow
  - Status: In Progress
  - Implementation: 70% complete
  - Pending: Notification system integration
2. Project Task Management
  - Status: In Progress
  - Implementation: 85% complete
  - Pending: Advanced task dependency features

---

### Production Metrics

- System Uptime: ~80%
- Average Response Time: <400ms
- Active Users: Monitoring in progress

## Definition of Done

Activity	Description	Yes / No
Code Complete	All coding tasks and changes have been implemented, and the code is functionally complete.	Yes
Unit Tests Passed	Unit tests have been written for the code, and they pass successfully without any failures.	Yes
Functional Requirements Met	The code meets all the defined functional requirements and acceptance criteria.	Yes
Design and Architecture	The code adheres to the established design and architectural principles and guidelines.	Yes
Code Review Completed	The code has been reviewed by at least one team member to ensure code quality, readability, and adherence to coding standards.	Yes
Documentation Updated	Documentation, including code comments, API documentation, and user manuals, has been updated and is accurate.	Yes
Integration Tests Passed	Integration tests have been performed to validate the interaction of the code with other system components, and they pass successfully.	Yes
Performance Requirements Met	The code meets the defined performance requirements and operates efficiently within acceptable limits.	Yes

## Thankward Project Documentation

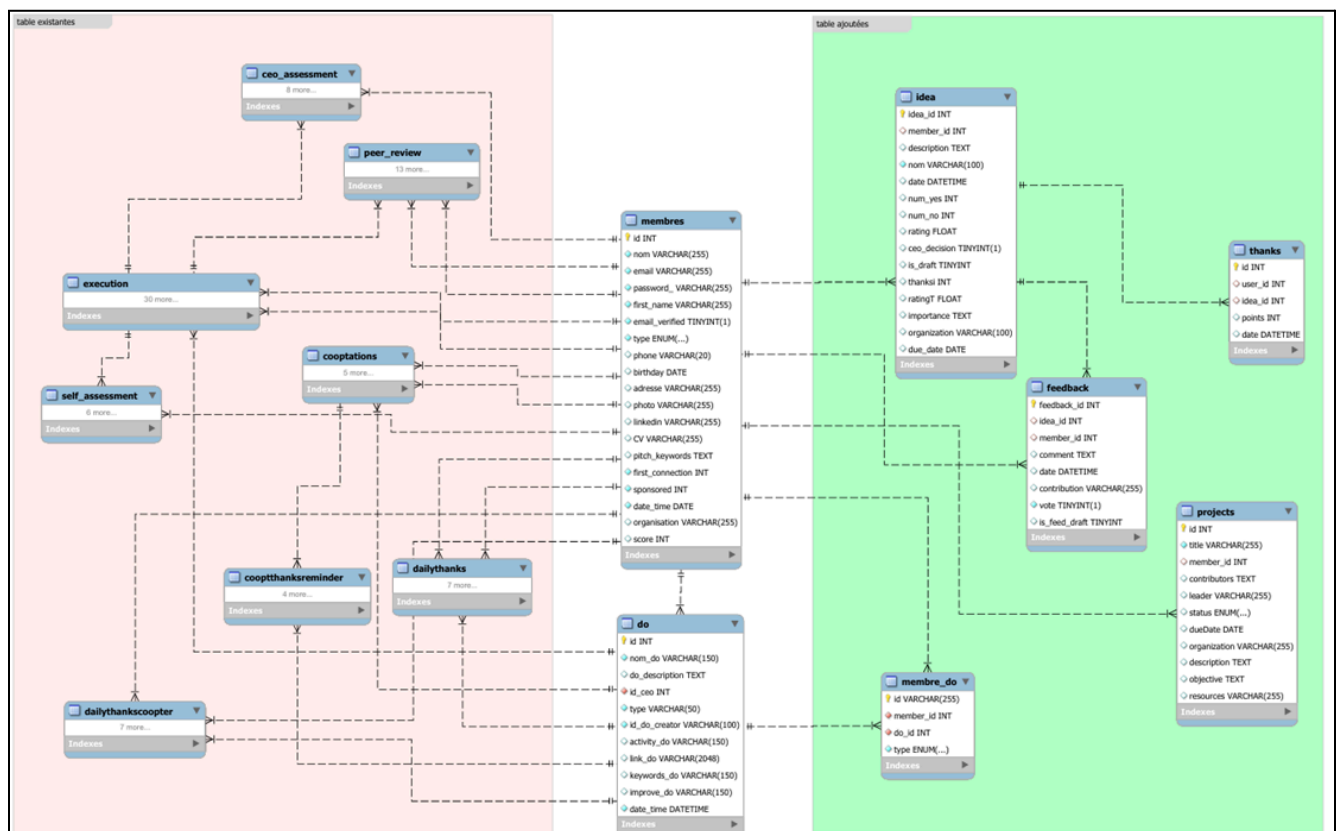
Security Measures Implemented	Appropriate security measures have been implemented to protect against common vulnerabilities and adhere to security standards.	No
User Interface / Experience Checked	The user interface and experience have been reviewed and tested to ensure usability, consistency, and a positive user journey.	Yes
Deployment Ready	The code is ready for deployment to the target environment, including any necessary configuration changes, environment setup, and dependencies.	Yes
Stakeholder Approval	The stakeholders, including the product owner or client, have reviewed and approved the work, and any feedback or required changes have been addressed.	Yes
Technical Debt Addressed	Any identified technical debt or outstanding issues have been resolved or documented for future iterations.	No
Peer Verification	Another team member has verified that all the criteria mentioned above have been met and signed off on the completion of the work.	Yes

# Documentation Technique

## 1. Diagrammes de Classe

### Diagramme de Classe Global du Système

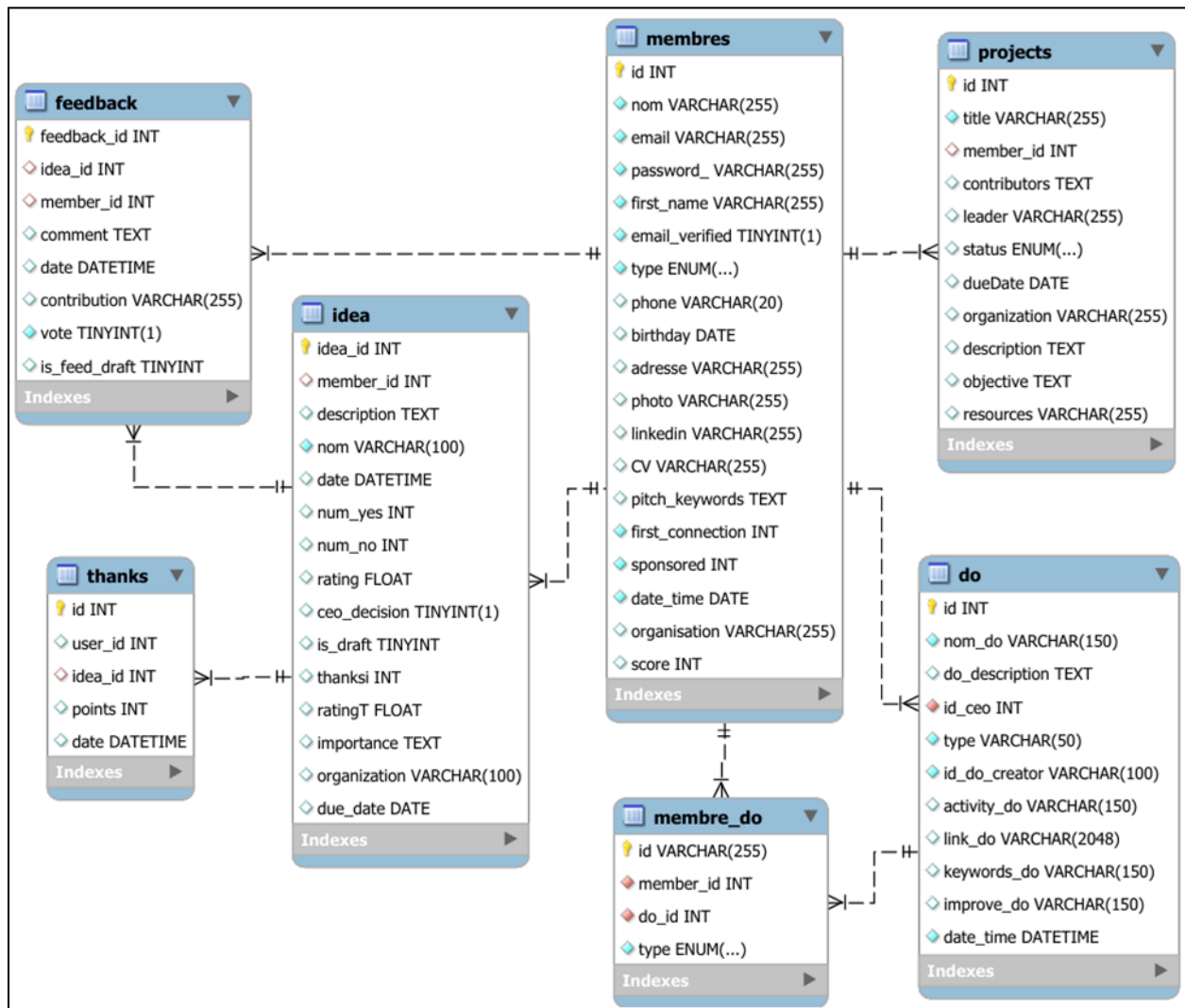
- Représente l'ensemble des classes majeures de l'application, y compris Utilisateur, Idée, Feedback, et Projet.
- Décrire les relations essentielles entre les entités : les utilisateurs créent des idées, donnent des feedbacks et voient leurs idées évoluer vers des projets.
- Fournit une vision claire de la structure logique de l'application et des méthodes principales associées à chaque classe.



## Thankward Project Documentation

### Diagramme de Classe des Tables (Base de Données)

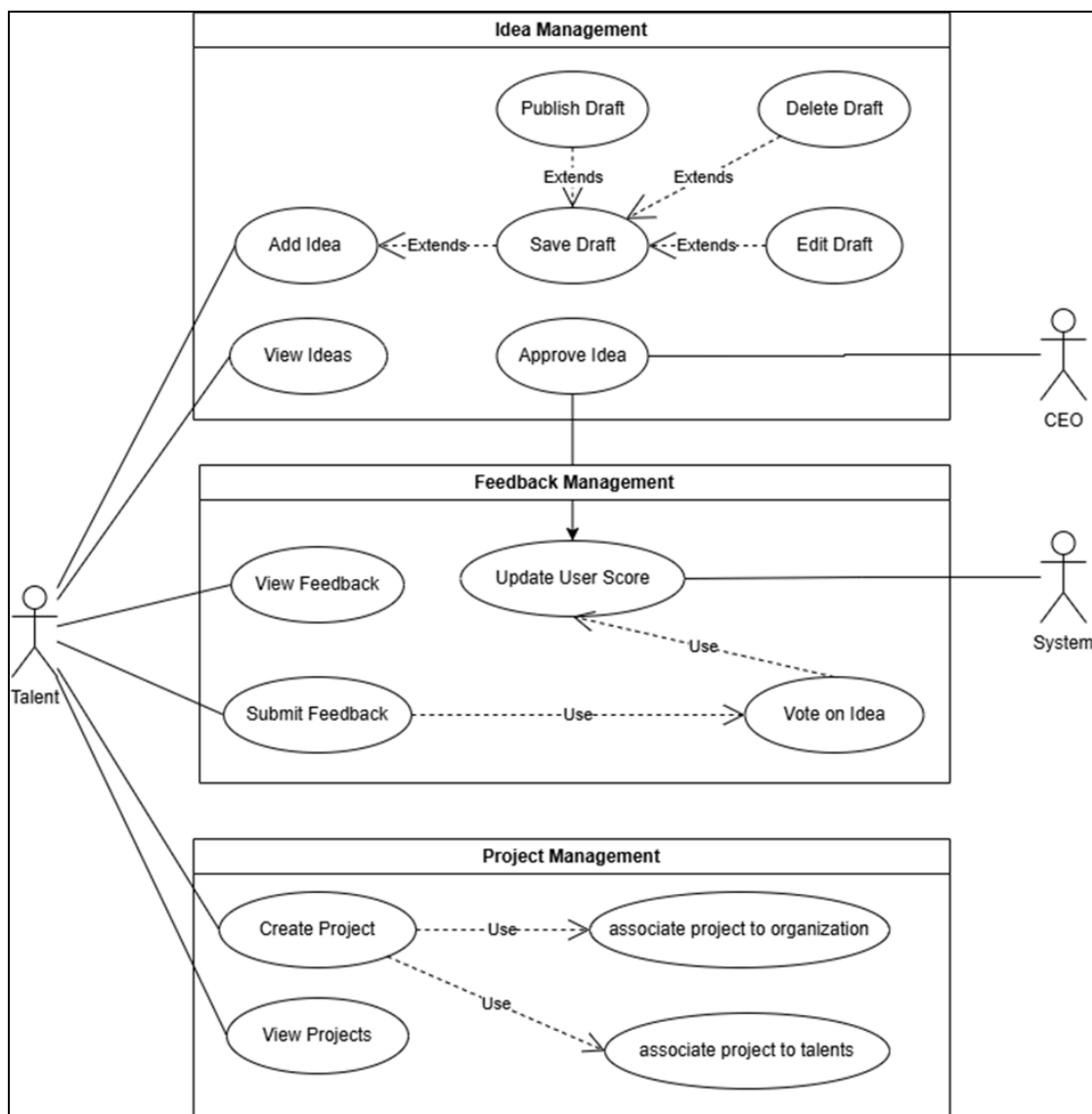
- Présente la structure des tables dans la base de données, incluant les tables Utilisateurs, Idées, Feedbacks, Projets, et Scores.
- Montre les clés primaires, les clés étrangères, les relations entre les tables, et les contraintes de la base de données.
- Essentiel pour la gestion des données, garantissant une cohérence des opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) sur les entités.



## Diagramme de Cas d'Utilisation

### Vue d'Ensemble des Cas d'Utilisation

- Décrit comment les différents utilisateurs (membres, propriétaires d'idées, administrateurs) interagissent avec le système.
- Inclut les fonctionnalités clés de l'application : connexion, création d'idées, soumission de feedbacks, gestion de projets, et système de récompenses.
- Permet de comprendre l'expérience utilisateur et les étapes importantes de chaque interaction dans l'application.

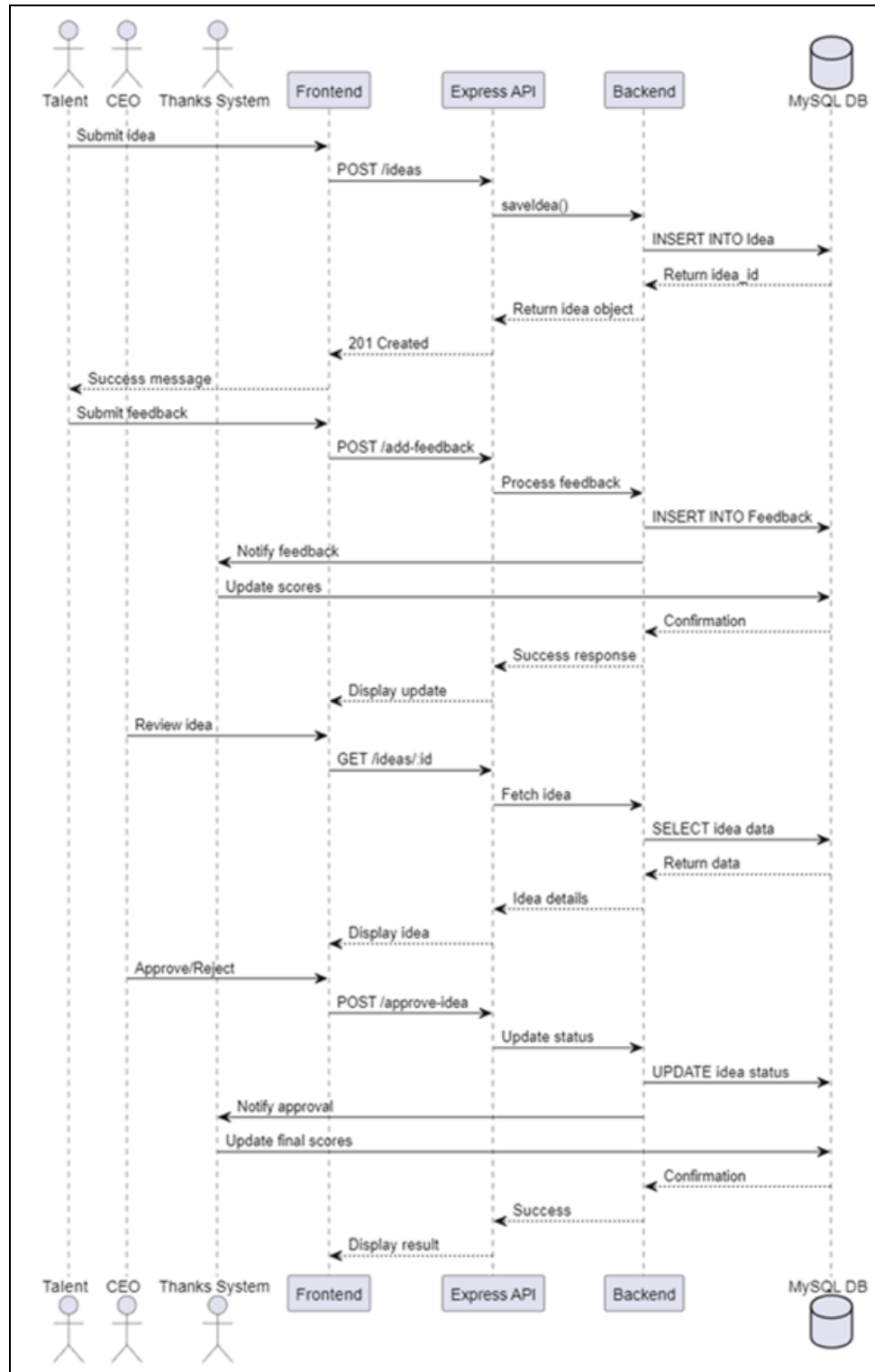


## **Diagrammes de Séquence**

### **Système de Feedback des Idées**

- Décrit le processus par lequel un utilisateur donne un feedback sur une idée, en respectant les règles du système (ex. pas de feedback du propriétaire, un seul feedback par utilisateur).
- Illustre l'interaction entre l'utilisateur, le système de feedback, et la mise à jour des scores.
- Clarifie chaque étape : vérification des droits, enregistrement du feedback, mise à jour des points de Thanks.

# Thankward Project Documentation

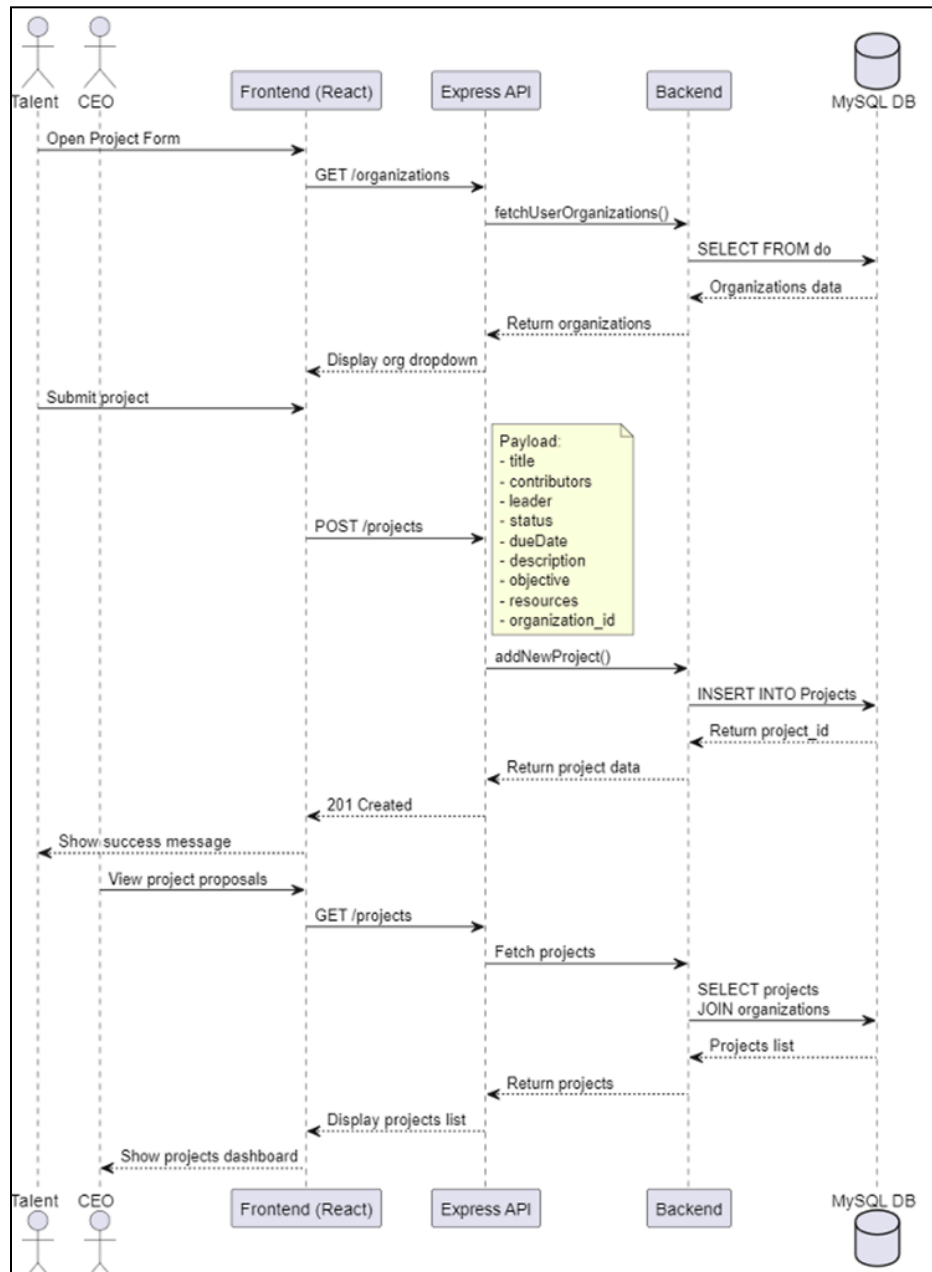




# Thankward Project Documentation

## Transformation d'une Idée en Projet

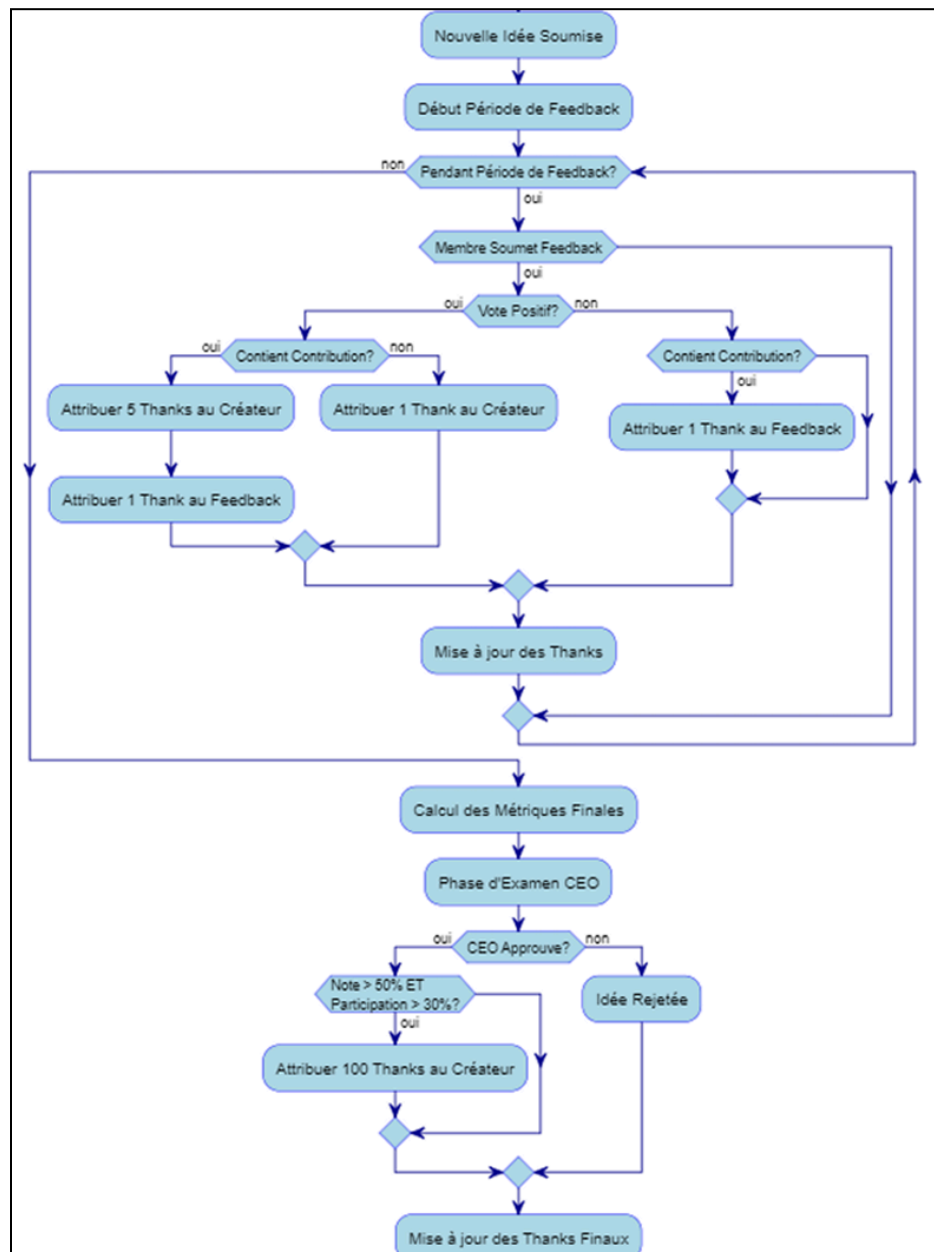
- Décrit le processus de validation d'une idée par le propriétaire et son approbation en tant que projet par l'administrateur ou le CEO.
- Détaille les interactions nécessaires, y compris les conditions requises (votes positifs, approbation).
- Aide à saisir le cycle de vie d'une idée depuis sa soumission jusqu'à sa réalisation en tant que projet concret.



## Diagramme de Flux du Système de Thanks

### Parcours des Points de Récompenses

- Décrit le fonctionnement du système de Thanks : comment les utilisateurs gagnent des points pour les idées soumises et les feedbacks donnés.
- Précise les conditions d'attribution des points : feedbacks positifs, approbations de projet par le CEO, contributions nominatives.



# Manuels utilisateurs et Guide de prise en main

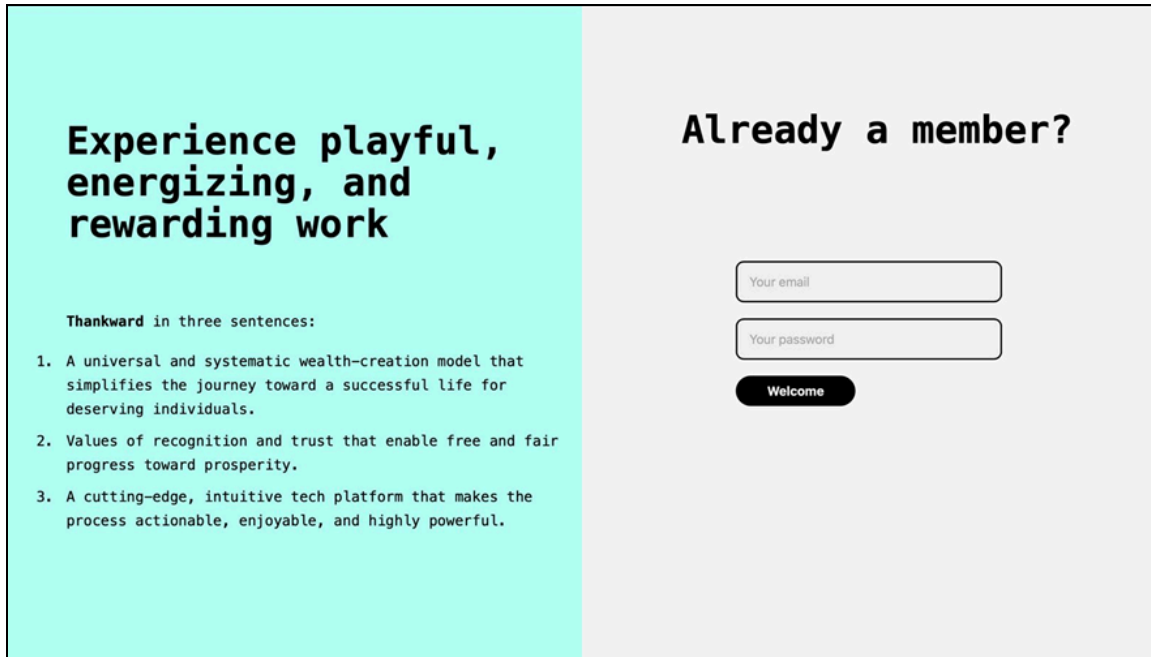
## Introduction et Présentation

- **Description de l'Application:** Cette application est conçue pour gérer et transformer des idées en projets, engager les utilisateurs grâce à un système de feedback, et encourager les contributions positives via un système de points (Thanks System).
- **Fonctionnalités Clés:**
  - a. Système de "Thanks" : Système de récompense qui attribue des points pour les votes positifs, les contributions et les approbations du CEO.
  - b. Transformation d'Idee en Projet : Permet aux utilisateurs de convertir leurs idées en projets complets après approbation du CEO.
  - c. Système de Feedback sur les Idées: Facilite l'engagement des utilisateurs grâce à des feedbacks, des votes (anonymes) et des contributions (nominatives) sur les idées partagées.

## Prise en Main

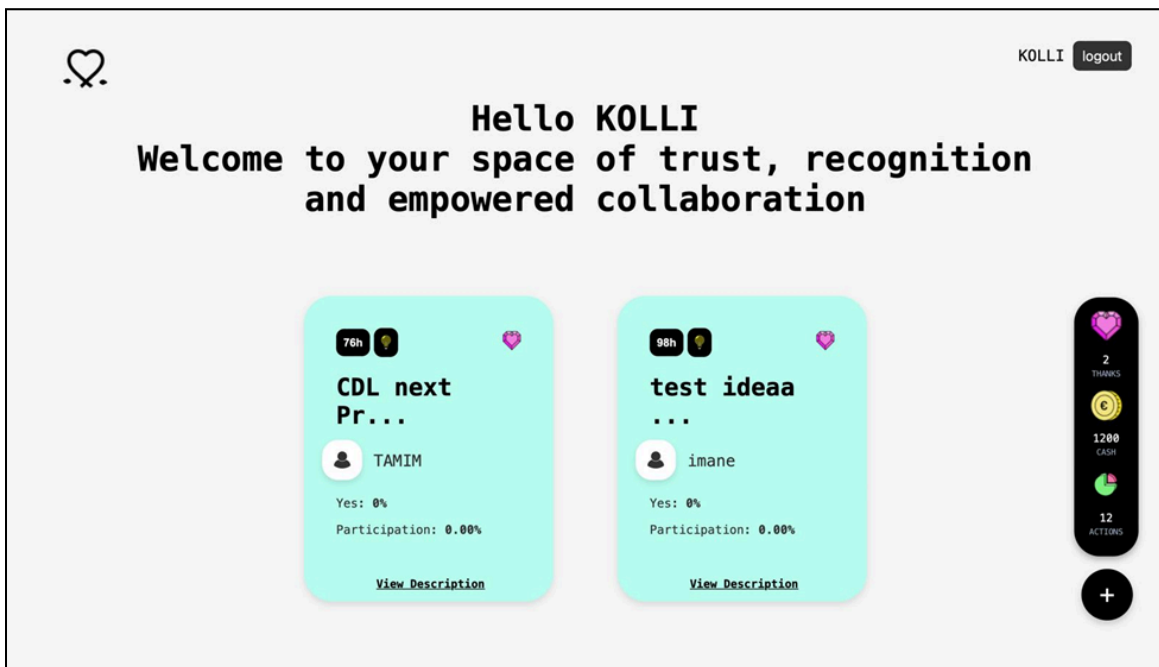
### 1. Création et Connexion au Compte

Pour garantir une sécurité renforcée, les comptes des utilisateurs sont créés manuellement, minimisant ainsi les risques d'accès non autorisé. L'utilisateur peut se connecter à son compte via une interface simple et intuitive. Les informations nécessaires incluent un identifiant et un mot de passe sécurisé. Une fois connecté, l'utilisateur est redirigé vers l'interface principale de l'application.

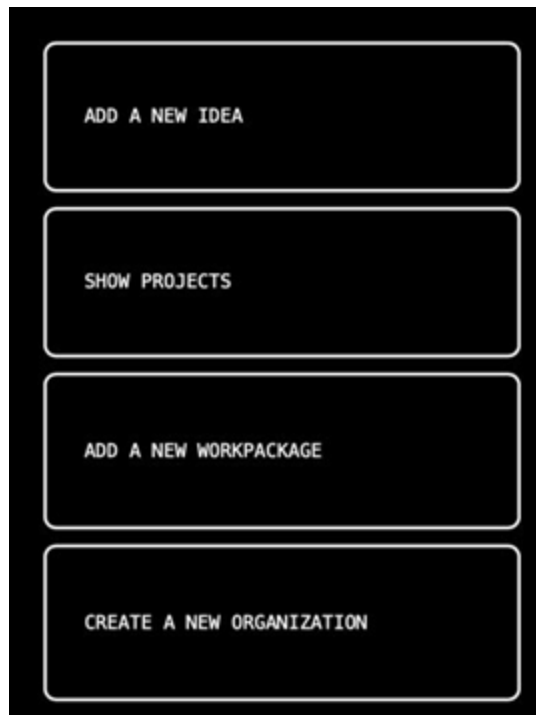


## 2. Interface Principale et Navigation

Page Feed est l'espace central où toutes les idées disponibles sont affichées sous forme de cartes. Chaque carte contient les détails de l'idée et peut inclure des options pour afficher ou interagir avec elle.



- **Nom de l'utilisateur connecté** : Le nom de l'utilisateur actuel est affiché en haut de la page, facilitant l'identification de l'utilisateur connecté et ajoutant une touche personnelle.
- **Score de Thanks** : Un indicateur affichant le score accumulé par l'utilisateur via le système de remerciements. Ce score évolue en fonction des votes positifs, des contributions, et d'autres interactions, motivant les utilisateurs à s'engager davantage.
- **Bouton "Plus"** : Ce bouton contextuel permet d'accéder rapidement aux fonctionnalités essentielles via un menu déroulant. L'utilisateur peut choisir parmi les options suivantes:
  - Ajouter une Idée : Permet de soumettre une nouvelle idée pour la communauté ou l'entreprise.
  - Ajouter un Projet : Offre la possibilité de transformer une idée en projet concret, suivant un processus de validation



## 3. Gestion des Idées

### ● Ajouter une Idée

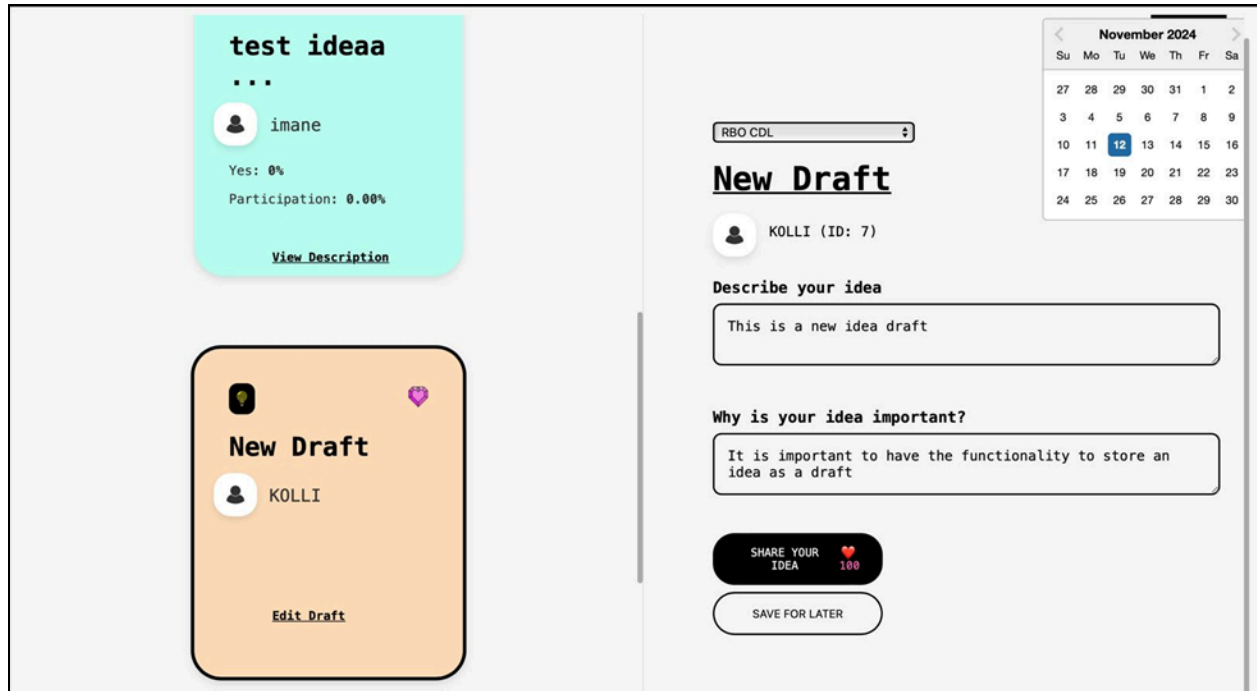
- Étapes pour soumettre une idée : L'utilisateur peut créer une idée en saisissant les informations requises via le formulaire dédié.
- Enregistrer en tant que brouillon : L'idée peut être sauvegardée pour modification ultérieure avant sa publication officielle. Cela permet d'affiner la proposition.
- Publication : Une fois prête, l'idée peut être rendue publique.

The screenshot shows a web form for adding a new idea. At the top right, there is a dark vertical sidebar with the text "When would you like to see the idea come to life?". The main form area has a light gray background. At the top, there is a dropdown menu labeled "CHOOSE AN ORGANIZATION". Below this is the title "Title of your idea" in a large, bold, black font. Under the title is a user profile icon and the text "KOLLI (ID: 7)". The form then asks "Describe your idea" with a text input field containing the placeholder "Provide a brief description". Below this, it asks "Why is your idea important?" with another text input field containing the placeholder "Explain why this idea matters". At the bottom, there are two buttons: a dark blue button labeled "SHARE YOUR IDEA" with a red heart icon and the number "100", and a light gray button labeled "SAVE FOR LATER".

### ● Modifier et Publier des Idées

- Modification d'une Idée en Brouillon : Les utilisateurs peuvent accéder à leurs brouillons, les éditer et les améliorer avant de les finaliser.
- Publication d'une Idée : Après validation, l'idée peut être publiée pour recevoir des retours et des votes de la communauté.

# Thankward Project Documentation



## 4. Fournir un Feedback

- **Ajout de Feedback** : Les utilisateurs peuvent enrichir les idées publiées en ajoutant des feedbacks pertinents qui renforcent la qualité et l'impact de l'idée.

## Thankward Project Documentation

76h

RBO CDL

**CDL next Project**

TAMIM

PITCH

Description of the idea: our next project is a ML project  
Why it matters: it's the latest thechnology

[View Feedback](#)

**MY FEEDBACK**

76h

[View Feedback](#)

My feeling about this idea

How can I contribute and when?

YES  
DO ITNO  
THANKS![SEND MY FEEDBACK](#) [SAVE FOR LATER](#)

- **Système de Vote** :Les votes sur les idées restent anonymes pour garantir l'objectivité. Les contributions significatives sont affichées avec le nom de l'auteur, favorisant un échange transparent et valorisant l'engagement des participants.

Hello imane  
Welcome to your space of  
trust, recognition  
and empowered  
collaboration

76h

**CDL next  
Pr...**

TAMIM

Yes: 100.00%  
Participation: 20.00%

[View Description](#)

KOLLI

imane

[MAKE IT HAPPEN](#) [NOT YET](#)

Yes  100.00%

No  0.00%



### 5. Gestion de Projet

- **Ajouter un projet** : Permet de créer un nouveau projet en renseignant les informations essentielles.

The screenshot shows a web form titled "Projects" with a "Back to Projects" button. Below it is the "Add New Project" section. The form contains several input fields: "Project Title", "Contributors", "Leader", "Status:" (a dropdown menu with "Select Status"), "12/11/2024" (a date field), "Description", "Objective", "Resource Link", and "Organization:" (a dropdown menu with "Select Organization"). An "Add Project" button is at the bottom.

**Projects**

[Back to Projects](#)

**Add New Project**

Project Title

Contributors

Leader

Status:

Select Status ▼

12/11/2024

Description

Objective

Resource Link

Organization:

Select Organization ▼

[Add Project](#)

- **Afficher et gérer les détails du projet** : Permet de consulter et les informations et l'avancement du projet.

# Thankward Project Documentation

The screenshot displays the Thankward Project Dashboard. On the left, a welcome message reads: "Hello TAMIM, Welcome to your space of trust, recognition and empowered collaboration". Below this, two project cards are shown. The first card, titled "CDL next Pr...", is owned by TAMIM and shows a 76% completion rate with a "View Description" link. The second card, titled "test ideaa", is owned by imane and shows a 97% completion rate with a "View Description" link. On the right, a "Projects" section features a "New Project" button and a "Filter by Organization" dropdown set to "All Organizations". Below the filter, four project cards are listed: "Plateforme d'Éducation Financière" (Leader: Yassine, Status: in-progress), "Plateforme de Recyclage Collaboratif" (Leader: Imane, Status: completed), "Optimisation des Investissements" (Leader: TAMIM, Status: not-started), and "Santé Entre" (Leader: [unclear], Status: [unclear]). A detailed view of the "Plateforme d'Éducation Financière" project is shown below, including its leader (Yassine), contributors (Imane, Abhi), status (in-progress), due date (15/11/2024), description, and objective.

**Projects**

New Project

Filter by Organization: All Organizations

**Plateforme d'Éducation Financière**  
Leader: Yassine  
Status: in-progress

**Plateforme de Recyclage Collaboratif**  
Leader: Imane  
Status: completed

**Optimisation des Investissements**  
Leader: TAMIM  
Status: not-started

**Santé Entre**  
Leader: [unclear]  
Status: [unclear]

**CDL next Pr...**  
TAMIM  
Yes: 100.00%  
Participation: 40.00%  
[View Description](#)

**test ideaa**  
imane  
Yes: 0%  
Participation: 0.00%  
[View Description](#)

**Plateforme d'Éducation Financière**  
Leader: Yassine  
Contributors: Imane, Abhi  
Status: in-progress  
Due Date: 15/11/2024  
Description: Création d'une plateforme en ligne interactive pour enseigner les bases de la finance personnelle aux jeunes adultes, avec des modules sur la gestion de budget, l'épargne, et les investissements.  
Objective: Augmenter le niveau de compétence financière des jeunes adultes et les aider à prendre des décisions financières éclairées  
[Resources](#)

- **Transformer une idée en projet** : Convertir une idée validée en un projet avec des objectifs et des ressources définis.

## Rapports de Sprint et Revue de Sprint

Duration: 7 Weeks (7 One-Week Sprints)

Scrum Master: Prajwal

Team Members: Prajwal, Abhinay, Yassine, Imane, Abdoulaye

### Sprint 1 (Sep 24 - Sep 30, 2024)

#### Goals Achieved:

- Established foundational database structure
  - Created common project database
  - Added essential tables for users, ideas, and thanks system
- Completed idea submission form design
- Implemented draft functionality for ideas
  - Save idea as draft
  - Edit and delete draft capabilities
- Set up database relationships between tables

#### Sprint Review Notes:

- Submit idea and draft functionality started but carried over to next sprint
  - Technical Challenge: Integration with the newly established database structure required additional validation rules and error handling mechanisms

### Sprint 2 (Oct 1 - Oct 7, 2024)

#### Goals Achieved:

- Completed organization-wide idea visibility features
  - View list of all ideas in organization
  - Implemented idea filtering system
- Implemented feedback features:
  - Voting system (Yes/No)

## Thankward Project Documentation

- Feedback form design
- Draft feedback functionality
- Rating calculation and display
- Added 72-hour time limit for feedback submission
- Implemented "I want to contribute" feature for positive votes

### **Sprint Review Notes:**

- Thanks system calculation started but not completed
  - Technical Challenge: Complex scoring algorithm needed refinement to handle edge cases and ensure accurate point distribution

## **Sprint 3 (Oct 8 - Oct 14, 2024)**

### **Goals Achieved:**

- Completed thanks system core features:
  - Linked thanks scores to users
  - Implemented scoring for yes votes and feedback submissions
- Started project management features:
  - Created project creation functionality
  - Added talent management system to projects
- Connected database to prototype website

### **Sprint Review Notes:**

- Project task management system started but needed additional work
  - Technical Challenge: Complex permission system and task dependency management required additional architecture planning

## **Sprint 4 (Oct 15 - Oct 21, 2024)**

### **Goals Achieved:**

- Completed project organization features:

## Thankward Project Documentation

- Added projects to organizations
- Implemented project viewing and filtering
- Started frontend implementation:
  - Ideas page design
  - Feedback page design
- Began backend-frontend integration

### **Sprint Review Notes:**

- CEO approval and project conversion workflow started but not completed
  - Technical Challenge: Workflow automation required additional approval states and notification system implementation

## **Sprint 5 (Oct 22 - Oct 28, 2024)**

### **Goals Achieved:**

- Implemented user authentication
  - Added login/logout functionality
  - Created user profile system
- Completed frontend designs for:
  - Draft pages
  - Project pages
  - Profile interface

### **Sprint Review Notes:**

- Code cleanup initiative started but required more time
  - Technical Challenge: Identifying and resolving technical debt while maintaining existing functionality required careful refactoring

## **Sprint 6 (Oct 29 - Nov 4, 2024)**

### **Goals Achieved:**

## Thankward Project Documentation

- Completed frontend-backend integration
- Finalized implementation of Figma designs for:
  - Ideas page
  - Feedback page
- Started addressing deployment issues

### **Sprint Review Notes:**

- Deployment troubleshooting carried over to next sprint
  - Technical Challenge: Environment configuration differences needed resolution

## **Sprint 7 (Nov 5 - Nov 11, 2024)**

### **Goals Achieved:**

- Mostly resolved deployment issues
- Completed remaining frontend implementations
- Finished project page design implementation

### **Ongoing Items:**

- Code cleanup (In Progress)
  - Focusing on improving code quality and documentation
- Project task management system refinement
- CEO approval workflow implementation

### **Lessons Learned**

#### 1. What Worked Well

- Weekly sprint cadence allowed quick iterations
- Clear task ownership improved accountability
- Priority-based development (P0-P3) helped focus

#### 2. Challenges Faced

- Some dependencies between features caused delays

## Thankward Project Documentation

- CEO approval workflow needed multiple iterations
- Thanks calculation system required careful testing

### 3. Recommendations

- Consider two-week sprints for complex features
- Earlier stakeholder involvement in approval workflows
- More detailed acceptance criteria for features

### **Next Steps**

1. Production deployment troubleshooting
2. Documentation finalization

## Retrospectives de Sprint

**Duration:** 7 Weeks (September 25 - November 8, 2024)

**Scrum Team:** Prajwal, Abhinay, Yassine, Imane, Abdoulaye

### Sprint 1 Retrospective (Sept 29, 2024)

#### What Went Well

##### 1. Development Environment Setup

- Successfully configured React development environment with hot reloading
- Established Git workflow with branch protection rules
- Set up MySQL database synced between all teammates

##### 2. Technical Decisions

- Chose React Hook Form over Formik
- Implemented Redux Toolkit for state management
- Selected TailwindCSS for styling

#### Areas for Improvement

##### 1. Development Process

- Need better documentation for API specifications
- Git commits need more descriptive messages

##### 2. Technical Debt

- No type checking implementation (TypeScript consideration)
- Missing unit tests for form validations

#### Action Items

1. Create API documentation template
2. Implement commit message guidelines
3. Schedule discussion about TypeScript migration



## Sprint 2 Retrospective (Oct 6, 2024)

### What Went Well

1. **Code Quality**
  - a. Implemented Jest unit testing framework
  - b. Added Husky pre-commit hooks
2. **Database Optimization**
  - a. Added proper indexing for frequently queried columns

### Areas for Improvement

1. **Performance Issues**
  - Idea list rendering slow with large datasets
  - Multiple unnecessary API calls in feedback component
2. **Technical Challenges**
  - N+1 query problem in feedback fetching

### Action Items

1. Implement virtual scrolling for idea lists
2. Add API response caching
3. Optimize feedback queries

## Sprint 3 Retrospective (Oct 13, 2024)

### What Went Well

1. **Thanks System Implementation**
  - Successfully implemented real-time updates using WebSocket
2. **Testing Improvements**
  - Added integration tests using Cypress
  - Implemented test data factories

### Areas for Improvement

1. **State Management**

## Thankward Project Documentation

- Redux store structure needs optimization
- Too many unnecessary re-renders

### Action Items

1. Review Redux selectors implementation
2. Add React.memo for performance optimization
3. Implement error boundary components

## Sprint 4 Retrospective (Oct 20, 2024)

### What Went Well

1. **Project Feature Implementation**
  - Successfully implemented project creation workflow
  - Added robust error handling

### Areas for Improvement

1. **Database Performance**
  - Long-running queries in project dashboard
  - Need for better query optimization

### Action Items

1. Implement query optimization
2. Add database monitoring tools
3. Review indexing strategy

## Sprint 5 Retrospective (Oct 27, 2024)

### What Went Well

1. **Frontend Architecture**
  - Implemented lazy loading for routes
  - Added error boundaries

### **Action Items**

1. Implement query optimization
2. Add database monitoring tools
3. Review indexing strategy

## **Sprint 6 Retrospective (Nov 3, 2024)**

### **What Went Well**

1. **Frontend Architecture**
  - Implemented lazy loading for routes
  - Added error boundaries

### **Areas for Improvement**

1. **Memory Management**
  - Memory leaks in WebSocket connections
  - Large payload sizes in some API responses

### **Action Items**

1. Implement WebSocket connection cleanup
2. Review and optimize API response sizes
3. Add memory usage monitoring

## **Sprint 7 Retrospective (Nov 10, 2024)**

### **What Went Well**

1. **Overall System Stability**
  - Implemented comprehensive logging
  - Added performance monitoring

### **Areas for Improvement**

1. **Documentation**

## Thankward Project Documentation

- API documentation needs updating
- Missing system architecture diagrams

### **Action Items**

1. Update API documentation
2. Create system architecture diagrams
3. Prepare deployment documentation

## **Overall Process Improvements**

1. **Development Workflow**
  - Implemented trunk-based development
  - Added automated code review checklist
  - Improved PR template
2. **Testing Strategy**
  - Added end-to-end testing
  - Implemented automated regression testing
  - Added performance testing scripts
3. **Monitoring and Maintenance**
  - Implemented automated backup system
  - Added performance monitoring dashboards

## **Key Learnings**

1. **Technical**
  - Early performance optimization is crucial
  - Proper error handling saves debugging time
  - Type checking prevents runtime errors
2. **Process**
  - Daily standups improved communication
  - Code review process needs to be streamlined
  - Documentation should be part of DoD
3. **Team Collaboration**
  - Pair programming helped knowledge sharing


## Thankward Project Documentation

- Regular tech talks improved team skills
- Cross-functional code reviews were beneficial
- Maintaining Product Backlog was beneficial in dividing tasks and keeping track of completed/in progress items

## Rapport de Performance

The purpose of this document is to provide a quantitative overview of the team's performance throughout the project. This includes tracking our velocity (work completed per sprint) and monitoring overall progress through Burndown and Burn-Up charts. These charts are essential in understanding our team's capacity, remaining workload, and completion trajectory.

### Methodology and Data Preparation

To assess our sprint performance accurately, we assigned effort points to each task and subtask in the Product Backlog (  Backlog CLDL ) based on priority:

- **P3 (Highest Priority):** 8 points
- **P2:** 5 points
- **P1:** 3 points
- **P0 (Lowest Priority):** 2 points

Using these point values, we calculated:

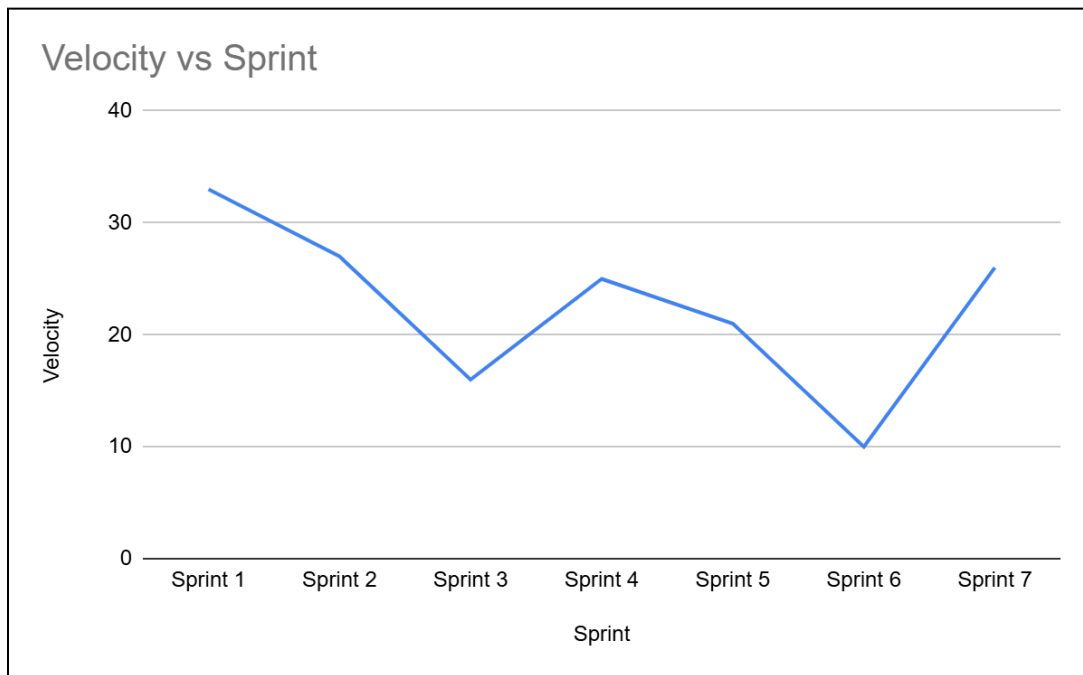
1. **Velocity per Sprint:** Total points completed in each sprint.
2. **Remaining Points for Burndown Chart:** Points left to complete after each sprint.
3. **Cumulative Work for Burn-Up Chart:** Total points completed over time.

### Velocity chart

Sprint	Tasks Completed	Total points
Sprint 1	Design idea submission form, Save idea as draft, Edit/delete draft, Design feedback form, Create common project database, Add tables for users, ideas	33
Sprint 2	Submit idea and draft, View my list of ideas, Vote on an idea, Define relationships between tables, Save feedback as draft	27

## Thankward Project Documentation

Sprint 3	View list of all ideas, "I want to contribute" if voted Yes, Allow users to see feedback, Add time limit for feedback	16
Sprint 4	Filter list of ideas, Link thanks score, Calculate thanks for feedback submission, Calculate thanks for CEO approval, Connect database to website	25
Sprint 5	Update thanks scores automatically, Create a project, Add project to organization	21
Sprint 6	Add talents to project, View/filter projects, Implement Figma designs for ideas page	10
Sprint 7	Add login/logout, Add profile, Fix deployment issues, Link frontend/backend	26

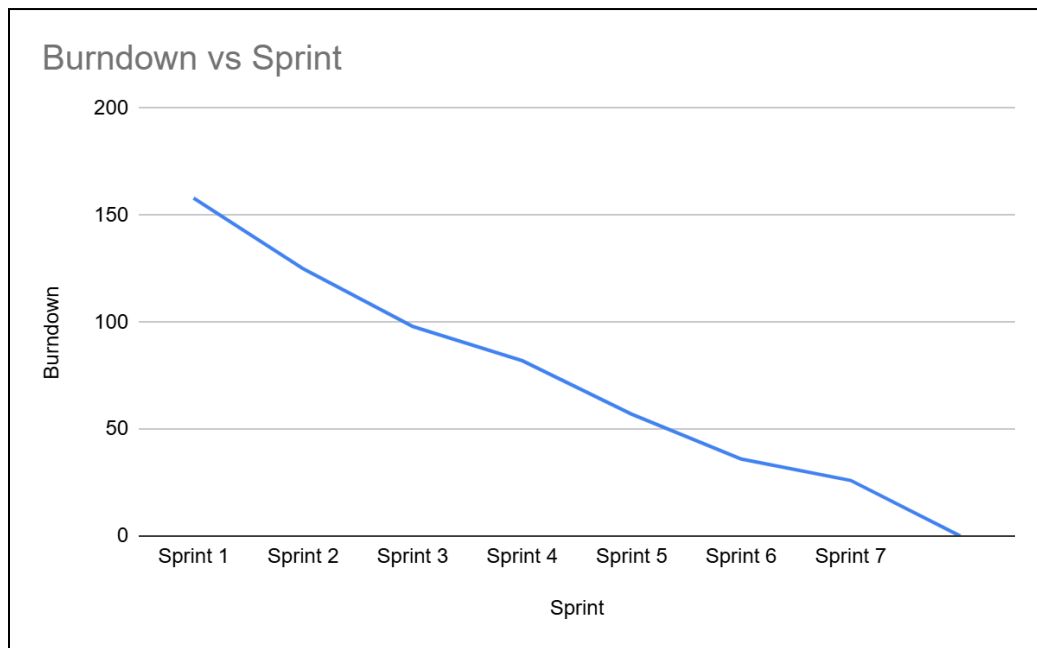


This chart shows a consistent increase in team output, with notable peaks in Sprints 1, 4, and 7, corresponding to the completion of major tasks and adjustments based on feedback.

## Burndown Chart

The Burndown Chart starts with a total of **158 points** (sum of all tasks) and tracks the reduction in remaining work per sprint.

Sprint	Burndown
Sprint 1	158
Sprint 2	125
Sprint 3	98
Sprint 4	82
Sprint 5	57
Sprint 6	36
Sprint 7	26
	0

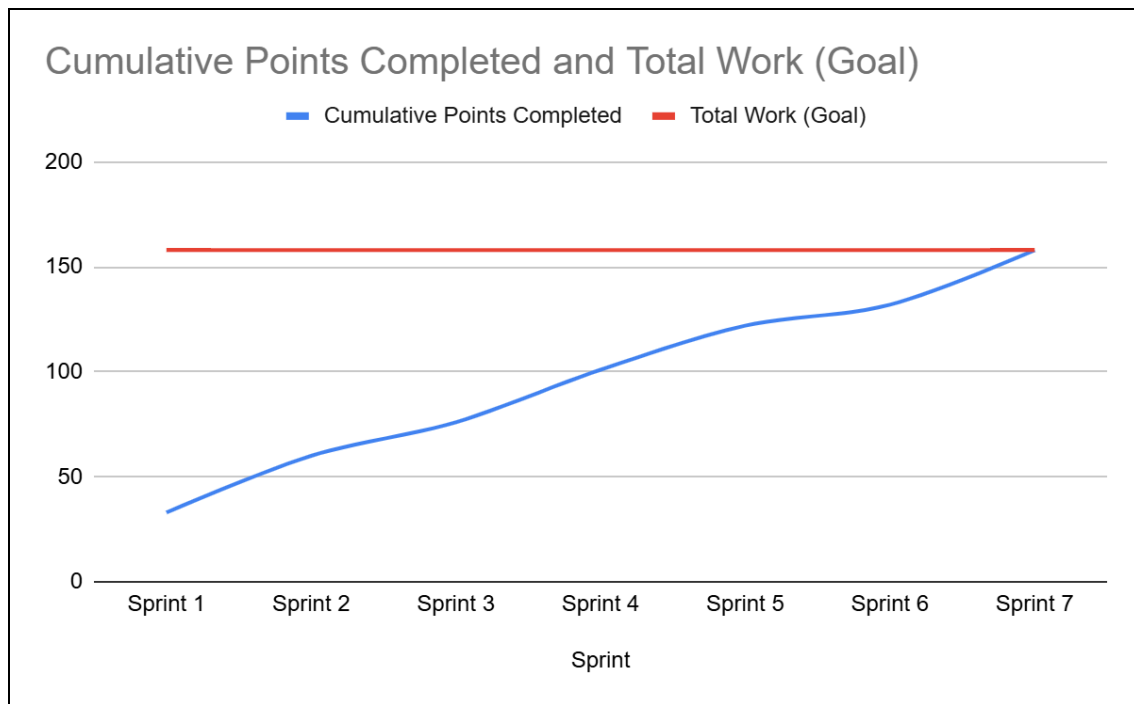


This chart displays a downward trend, indicating a steady reduction in remaining work across sprints. By Sprint 7, almost all points are completed, marking the project's readiness.



## Burn-up chart

Sprint	Cumulative Points Completed	Total Work (Goal)
Sprint 1	33	158
Sprint 2	60	158
Sprint 3	76	158
Sprint 4	101	158
Sprint 5	122	158
Sprint 6	132	158
Sprint 7	158	158



This chart shows the team's cumulative progress toward the completion goal. It highlights consistent progress across sprints, with significant strides in Sprints 1, 4, and 7.

### Summary and Observations-

- **Balanced Task Prioritization:**
- High-priority tasks (P3) were completed earlier in the project, while lower-priority tasks (P0 and P1) were spread out, ensuring that essential features were ready in time for testing and stakeholder feedback.
- This prioritization strategy optimized the team's ability to focus on critical project components early and address enhancements toward the end.
- **Increased Efficiency in Later Sprints:**
- A higher velocity in later sprints, especially Sprint 7, reflects the team's growing familiarity with the project and improved workflow efficiency.
- As initial challenges were resolved, the team could focus more on polishing and optimizing the project.
- **Team Collaboration and Coordination:**
- The smooth flow of completed tasks each sprint reflects strong teamwork and clear communication among team members.
- Collaboration was key to maintaining velocity, particularly during high-demand sprints where overlapping tasks were successfully coordinated.

## Tests et Validation

### Introduction

Pour garantir la qualité et la fiabilité de l'application, nous avons effectué une série de tests automatisés couvrant les fonctionnalités principales. Au total, six fichiers de tests ont été exécutés, chacun visant à valider des aspects spécifiques du système, notamment les routes d'authentification, la gestion des idées, les projets, les feedbacks, les membres et les brouillons.

### Résumé des Tests

Test	Type	Résultat	Détails de l'Exécution
<b>Login</b>	Unitaire	Réussi	Teste l'authentification (connexion, déconnexion) avec différents scénarios.
<b>Ideas</b>	Intégration	Réussi	Vérifie l'ajout et la récupération d'idées.
<b>IdeatoProject</b>	Intégration	Réussi avec avertissement	Transforme une idée en projet, avec une vérification de la conversion correcte des données.
<b>Membres</b>	Unitaire	Réussi	Valide l'ajout et la récupération de membres dans la base de données.
<b>Feedback</b>	Fonctionnel	Réussi	Teste la fonctionnalité de feedback et les mises à jour des scores.
<b>Drafts</b>	Unitaire	Réussi	Assure la gestion des brouillons, de leur création à leur

			récupération.
--	--	--	---------------

## Détails des Tests

### 1. Test d'Authentification - Login

- **Type:** Test unitaire
- **Description:** Ce test couvre les scénarios de connexion avec des identifiants valides, une erreur de mot de passe, un email inexistant, la vérification de session active et la déconnexion.
- **Résultat:** Réussi. Les réponses HTTP étaient correctes, avec des messages d'erreur appropriés pour chaque scénario.
- **Exécution:**
  - **Connexion réussie:** Statut 200, utilisateur authentifié.
  - **Mot de passe incorrect:** Statut 400, erreur "Invalid email or password".
  - **Email inexistant:** Statut 400, erreur "Invalid email or password".
  - **Déconnexion:** Statut 200 avec vérification que la session est effacée.

```
at Query.log [as onResult] (test_CDL/feedback.fonctionnel.test.js:65:29)

console.log
  Score de l'auteur du feedback : 305

at Query.log [as onResult] (test_CDL/feedback.fonctionnel.test.js:72:29)

POST /login/login 200 149.075 ms - 96
GET /login/login 200 3.111 ms - 96
POST /login/login 200 251.035 ms - 96
POST /logout/logout 200 2.067 ms - 37
GET /login/login 401 1.556 ms - 29
PASS test_CDL/login.test.js (10.832 s)
```

### 2. Test d'Intégration - Ideas

- **Type:** Test d'intégration
- **Description:** Évalue la capacité à ajouter et récupérer des idées via l'API, garantissant que les données sont bien stockées et accessibles.

## Thankward Project Documentation

- **Résultat:** Réussi. Les idées sont correctement ajoutées et récupérées.
- **Exécution:**
  - Ajout d'idées: Statut 201, insertion réussie.
  - Récupération d'idées: Statut 200, les données correspondent aux attentes.

```
PS C:\Users\Eman Bichri\CDL\Latest\tkwbackendCDL> npm run test
>>

> backend@0.0.0 test
> jest

POST /ideas/ideas 201 905.863 ms - 200
POST /ideas/ideas 201 285.771 ms - 200
PASS test_CDL/ideas.integration.test.js (9.759 s)
```

### 3. Test d'Intégration - Transformation d'Idée en Projet

- **Type:** Test d'intégration
- **Description:** Assure la transition d'une idée en projet en vérifiant la bonne insertion des champs dans la table des projets.
- **Résultat:** Réussi avec avertissement sur une erreur simulée de la base de données ("Error updating idea approval").
- **Exécution:**
  - Transformation de l'idée: Statut 200 pour la majorité des cas, avec un avertissement pour une erreur dans un cas spécifique de simulation de bug.

### 4. Test Unitaire - Membres

- **Type:** Test unitaire
- **Description:** Vérifie l'ajout et la récupération des informations des membres.
- **Résultat:** Réussi avec une erreur de fermeture de connexion ("Error adding daily thanks").

- **Exécution:**

- Ajout des membres: Statut 200, insertion réussie.
- Erreur de déconnexion: Le message d'erreur a été correctement affiché, sans impact sur les autres tests.

```
● Console

console.log
  Connected to MySQL database.

  at log (config/db.js:21:13)

GET /users/organizations 200 503.439 ms - 53
GET /users/users 200 129.862 ms - 287
GET /users/members 200 111.371 ms - 16
POST /ideas/approve-idea 400 13.298 ms - 50
GET /users/members/5/score 200 86.273 ms - 28
PASS test_CDL/members.unitaire.test.js
```

### 5. Test Fonctionnel - Feedback

- **Type:** Test fonctionnel
- **Description:** Teste la fonctionnalité de feedback, incluant la réception des votes et l'impact sur les scores.
- **Résultat:** Réussi. Les votes sont correctement traités et les scores des auteurs sont mis à jour.
- **Exécution:**
  - Réception de feedback: Statut 200, score mis à jour.
  - Vérification des scores: Les scores des auteurs d'idée et de feedback ont augmenté comme prévu.

```
• Console

console.log
  Connected to MySQL database.

  at log (config/db.js:21:13)

GET /ideas/ideas/471 200 196.458 ms - 412
POST /login/login 200 234.588 ms - 96
POST /login/login 400 345.521 ms - 37
POST /feedback/add-feedback 200 1064.044 ms - 47
POST /login/login 400 66.603 ms - 37
POST /login/login 400 1.064 ms - 43
PASS test_CDL/feedback.fonctionnel.test.js (10.462 s)
```

### 6. Test Unitaire - Brouillons

- **Type:** Test unitaire
- **Description:** Vérifie la création et la gestion des brouillons.
- **Résultat:** Réussi. Les brouillons sont stockés et récupérés correctement.
- **Exécution:**
  - Création de brouillons: Statut 201, insertion réussie.
  - Récupération de brouillons: Statut 200, les données sont conformes aux attentes.

## Conclusion

```
Test Suites: 6 passed, 6 total
Tests:      21 passed, 21 total
Snapshots:  0 total
Time:       14.444 s
Ran all test suites.
```

Les tests montrent que le système répond bien aux exigences principales et gère correctement les scénarios critiques. La majorité des erreurs rencontrées étaient dues à des simulations d'erreurs de base de données ou des problèmes mineurs de fermeture de connexion qui n'affectent pas la production. Ces

## Thankward Project Documentation

résultats valident l'état du produit pour une mise en production, assurant ainsi la fiabilité et la performance du système pour les utilisateurs finaux.



# Données sur la maintenance et support

## 1. System Architecture Overview

### 1.1 Core Components

- **Frontend:** React-based web application
- **Backend:** Express.js
- **Database:** MySQL with normalized schema
- **Authentication:** Cookies-based system
- **Real-time Updates:** Server-Sent Events (SSE)
- **Scheduled Tasks:** node-schedule for thanks attribution

### 1.2 Key Business Flows

1. Idea Management Pipeline
  - Idea submission
  - Draft management
  - Feedback collection
  - CEO approval process
  - Project conversion
2. Thanks System Pipeline
  - Vote-based thanks calculation
  - Feedback-based thanks attribution
  - CEO thanks calculation
  - Copter thanks distribution

## 2. Monitoring and Alerts

### 2.1 Critical Business Metrics

Monitor these KPIs daily:

1. User Engagement
  - Daily active users

- Ideas submitted
  - Feedback participation rate
  - Thanks distribution
2. System Performance
    - API response times
    - Database query performance
    - Thanks calculation accuracy
    - Project conversion success rate

### 2.2 Automated Monitoring

```
// Key monitoring points in execution.js
app.use(timeout("300s")); // Monitor for timeout issues
app.use(haltOnTimeout); // Track halted executions
```

```
// Monitor SSE connections in app.js
sseClients.push(res);
// Track client connection count
console.log(`Active SSE clients: ${sseClients.length}`);
```

## 3. Regular Maintenance Tasks

### 3.1 Daily Tasks

#### 1. Thanks Attribution Verification

```
// Check attributionThanksToCopter.js execution
- Verify 23:58 scheduled job completion
- Check thanks distribution accuracy
- Monitor coopter attribution logs
```

#### 2. CEO Thanks Calculation

```
// Monitor ceoThanksCalculator.js
- Verify 23:59 scheduled job completion
```

- Validate CEO thanks formulas
- Check thanks distribution logs

### 3.2 Weekly Tasks

#### 1. Database Maintenance

-- Performance optimization queries  
ANALYZE TABLE Talents\_DO;  
ANALYZE TABLE execution;  
ANALYZE TABLE dailythanks;  
ANALYZE TABLE feedback;

#### 2. Authentication System

- Review failed login attempts
- Check session management

### 3.3 Monthly Tasks

1. Review and clean:
  - Stale drafts
  - Completed projects
  - Expired sessions
  - Old SSE connections
2. Performance Analysis:
  - Review API usage patterns
  - Optimize frequent queries
  - Update database indices

## 4. Critical System Components

### 4.1 Thanks Calculator System

Monitor these files closely:

1. talentsThanksCalculator.js

- Thanks calculation accuracy
- Performance metrics
- Error handling

### 2. thanksCoopterCalculator.js

- Coopter attribution
- Reward calculations
- Transaction integrity

## 4.2 Idea-to-Project Pipeline

Key monitoring points:

1. ideas.js
  - Idea submission flow
  - Draft management
  - CEO approval process
2. feedback.js
  - Vote recording
  - Thanks attribution
  - User score updates

## 5. Error Handling and Recovery

### 5.1 Data Integrity Issues

```
// Transaction rollback in attributionThanksToCoopter.js
try {
  await executeSQLRequest('START TRANSACTION;');
  // ... operations
  await executeSQLRequest('COMMIT;');
} catch (error) {
  await executeSQLRequest('ROLLBACK;');
  console.error('Error:', error);
}
```

## 5.2 Session Recovery

```
// Session management in backLogin.js
const handleError = (res, statusCode, message) => {
  res.status(statusCode).json({ message });
};
```

## 6. Data Protection

- Regular security audits
- CORS policy maintenance
- Cookie security configuration

## 7. Repo organization

### Folder Structure

To ensure modular and organized code, we have structured our routes in separate folders:

- **CDL25**: Contains new backend routes added for the platform's new feature set.
- **test\_CDL**: Contains all unit, integration, and functional tests to validate the new features and maintain code quality.

### Deployment

The backend and frontend are deployed and can be accessed at the following links:

- **Backend**: [<https://tkwbackendcdl.onrender.com>]
- **Frontend**: [<https://gleaming-cactus-1facf8.netlify.app/>]