

جامعة أبو بكر بلقايد
UNIVERSITY OF TLEMCCEN



People's Democratic Republic of Algeria

Abou Bekr Belkaid University - Tlemcen

College of Sciences

Department of Computer Science

Final study thesis for obtaining a Master's degree in Computer Science

Option: Software Engineering

Subject

**Developing an AI-Based Approach for Project
Requirements Analysis and Management**

Prepared by:

Zerrouk Mohammed Hacene

Presented on June 29, 2025 before the jury composed of

Mr. Salim Ziani-Cherif

President (University of Tlemcen)

Mr. Amine Brikci Nigassa

Examiner (University of Tlemcen)

Mr. Azeddine Chikh

Supervisor (University of Tlemcen)

Academic Year: 2024 - 2025

Table of contents

I. General Introduction.....	1
I.1 Introduction.....	2
I.2 Project context.....	2
I.3 Thesis organization.....	2
II. Background.....	4
II.1 Introduction.....	5
II.2 Agile history and basics.....	5
II.3 Scrum history and basics.....	7
II.3.1 The origin of scrum.....	7
II.3.2 Scrum values.....	8
II.3.3 Scrum roles.....	9
II.3.3 Scrum processes.....	12
II.4 Estimation in project management.....	13
II.4.1 Project estimation.....	13
II.4.2 Importance of estimating a project.....	13
II.4.3 Improving project planning through accurate estimation.....	13
II.4.4 Agile story points estimation.....	14
II.5 Conclusion.....	14
III. Problem statement.....	16
III.1 Introduction.....	17
III.2 Challenges of incomplete or hidden requirements.....	17
III.3 Case studies.....	18
III.4 Analysis of delays due to hidden or incomplete requirements.....	18
III.5 Conclusion.....	19
IV. Proposed solution: implementation.....	20
IV.1 Introduction.....	21
IV.2 Project conception.....	21
IV.2.1 Use-case diagram.....	22
IV.2.2 Sequence diagrams.....	26
IV.2.2.1 Automatic generation of user stories based on epic and project context.....	26
IV.2.2.2 AI-Assisted developer consultation to refine user stories.....	28
IV.2.2.3 Automatic estimation based on 3 historical user stories.....	30
IV.2.3 Entity-relationship diagram.....	32
IV.2.3.1 Users and roles (Section 1).....	32
IV.2.3.2 Domain, platform and technologie (DPT) (Section 2).....	32

IV.2.3.3 Epics and user stories (Section 3).....	33
IV.2.3.4 Tasks (Section 4).....	33
IV.3 Detailed explanation.....	34
IV.3.1 Users and roles.....	34
IV.3.2 Domain, platform and technologie.....	35
IV.3.3 Epic and user story.....	36
IV.3.4 Tasks and acceptance criteria.....	37
IV.4 Tools and technologies used.....	38
IV.5 Project visual overview.....	40
General conclusion.....	46
References.....	47
Abstract.....	50

Table of figures

FIGURE 1 - History of scrum [9].....	7
FIGURE 2 - Scrum values [10].....	8
FIGURE 3 - Scrum team [10].....	9
FIGURE 4 - Scrum process [10].....	12
FIGURE 5 -Naming the pain in requirements engineering contemporary problems,causes, and effects in practice [11].....	17
FIGURE 7 - Use-Case Diagram.....	23
FIGURE 8 - Automatic generation of user stories based on epic and project context.....	27
FIGURE 9 - AI-Assisted developer consultation to refine user stories.....	29
FIGURE 10 - Automatic estimation based on 3 historical user stories.....	31
FIGURE 11 - high-level entity-relationship diagram.....	32
FIGURE 12 - Users and roles.....	34
FIGURE 13 - Domain, platform and technologie.....	35
FIGURE 14 - Epic and user story.....	36
FIGURE 15 - Tasks.....	37
FIGURE 16 - Tools and technologies used.....	38
FIGURE 17 - Dashboard.....	40
FIGURE 18 - Projects.....	40
FIGURE 19 - Epics.....	41
FIGURE 20 - User Stories.....	41
FIGURE 21 - Tasks.....	42
FIGURE 23 - Domains.....	42
FIGURE 24 - Platform.....	43
FIGURE 25 - Technologies.....	43
FIGURE 26 - Users.....	44
FIGURE 27 - Roles.....	44
FIGURE 29 - Notifications.....	45

Acknowledgments

First and foremost, I am deeply grateful to Allah for granting me the strength, patience, and guidance to complete this journey. Without His help, none of this would have been possible.

I would like to express my sincere gratitude to Mr. Azeddine Chikh, my supervisor, for his invaluable guidance, continuous support, and constructive feedback throughout the development of this project. His supervision has played a key role in shaping both the content and direction of this work.

I am also deeply thankful to Mr. Salim Ziani-Cherif, who will graciously serve as President of the jury. His guidance and encouragement during the early stages of this project were especially impactful and helped establish a clear and confident foundation for the work.

I also extend my appreciation to Mr. Amine Brikci Nigassa, who will serve as Examiner, for his participation and forthcoming evaluation of this project.

My sincere thanks also go to the FiveAngles team, especially to Mr. Mohammed AlAloul, CEO, for his support and for providing key technical resources that contributed to the success of this project.

I would also like to thank the Atqin community, a group of supportive friends whose encouragement, presence, and sincere advice helped me stay focused and motivated throughout this journey.

Finally, I would like to express my gratitude to the University of Abu Bekr Belkaid for providing a solid academic environment and the institutional support necessary for the successful completion of this work.

Dedication

To my beloved parents and family,

Your unwavering support, constant prayers, and unconditional love have been the foundation of my strength and perseverance throughout this journey.

This work is a reflection of your sacrifices and encouragement. I dedicate it to you with deepest gratitude and love.

I. General Introduction

I.1 Introduction.....	2
I.2 Project context.....	2
I.3 Thesis organization.....	2

I.1 Introduction

We are developing an AI-powered Requirement Management Application designed to enhance project planning and execution. By leveraging artificial intelligence, the platform delivers intelligent estimations, improving accuracy and efficiency throughout the project lifecycle. It automates the generation of epics, user stories, and tasks, streamlining requirement management and enabling teams to make data-driven decisions, optimize workflows, and achieve successful project outcomes.

I.2 Project context

Five Angles is a software development company with a track record of delivering tailored digital solutions across various industries. Over the years, the company has faced recurring challenges in the area of requirement management, particularly in capturing, interpreting, and maintaining alignment on project requirements throughout the development lifecycle. These challenges have often led to miscommunication, scope creep, and delays in project delivery.

Motivated by the need to enhance clarity and efficiency in requirement handling, Five Angles is exploring the integration of artificial intelligence to support and optimize this critical phase of software development. The proposed AI-driven solution aims to reduce ambiguity, improve stakeholder collaboration, and ultimately lead to better project outcomes through smarter requirement management.

I.3 Thesis organization

This thesis is structured into several chapters to provide a comprehensive overview of the research, development, and implementation of the AI-powered Requirement Management Application.

The **first chapter** provides essential background information. It begins with a brief introduction to agile methodologies, then explores the history and fundamentals of Agile and Scrum. It also discusses estimation practices within project management, setting the groundwork for understanding the problem space this project addresses.

The **second chapter** defines the problem in detail. It focuses on the recurring issues faced by software development teams, particularly incomplete or hidden requirements. Drawing on real case studies from Five Angles, this chapter highlights how these issues negatively impact project outcomes and reinforce the need for a smarter solution.

The **third chapter** presents the proposed solution and its implementation. It describes the system's conception, supported by use-case diagrams, sequence diagrams, and a data model. It also outlines the tools and technologies used, including Laravel, React, Inertia.js, OpenAI, and Filament PHP. The system architecture is detailed through the use of the Action Class design pattern and the MVC framework.

Finally, the **thesis concludes** with a general summary of the project outcomes, its impact, and future directions for improvement or expansion. References are included at the end to acknowledge the sources that informed this work.

II. Background

II.1 Introduction.....	5
II.2 Agile history and basics.....	5
II.3 Scrum history and basics.....	7
II.3.1 The origin of scrum.....	7
II.3.2 Scrum values.....	8
II.3.3 Scrum roles.....	9
II.3.3 Scrum processes.....	12
II.4 Estimation in project management.....	13
II.4.1 Project estimation.....	13
II.4.2 Importance of estimating a project.....	13
II.4.3 Improving project planning through accurate estimation.....	13
II.4.4 Agile story points estimation.....	14
II.5 Conclusion.....	14

II.1 Introduction

A thorough understanding of foundational concepts in Agile methodologies, Scrum practices, and project estimation techniques is essential for appreciating the context and significance of the proposed AI-powered Requirement Management Application.

These methodologies form the backbone of modern software development, emphasizing iterative progress, team collaboration, and adaptability to change.

Estimation, in particular, plays a critical role in planning and resource allocation, yet remains one of the most challenging aspects due to its dependency on accurate and complete requirements.

This chapter explores the historical development and core principles of Agile and Scrum, followed by a discussion on estimation practices in project management. By establishing this theoretical foundation, we provide the necessary context to better understand the problems identified and how our proposed solution seeks to address them.

II.2 Agile history and basics

In **February 2001**, seventeen software developers gathered at **Snowbird, Utah**, to discuss alternatives to traditional, rigid software development methodologies. Despite differing backgrounds, ranging from **Extreme Programming** and **SCRUM** to **Crystal** and **DSDM**, they found common ground and created the **Agile Manifesto**, a set of guiding values and principles for software development.

They named themselves the **Agile Alliance**, aiming to promote collaboration, adaptability, and people-centric approaches over bureaucracy and excessive documentation.

The manifesto wasn't just about techniques like **pair programming** or **refactoring**, but about deeper **values**, respect, and creating humane, productive work environments. It was a response to the frustration with corporate structures that prioritized processes over people and real results.

The movement emphasized:

- Simplicity in planning and documentation.
- Prioritizing working software over excessive paperwork.
- Embracing change and uncertainty.
- Putting people and collaboration at the center.

The meeting at Snowbird followed earlier conversations and meetups (like the one at Rogue River Lodge in 2000), where dissatisfaction with the term "Lightweight Methodologies" was growing. The term "Agile" emerged as a better fit, though not everyone initially liked or knew how to pronounce it.

Bob Martin, Kent Beck, Martin Fowler, Alistair Cockburn, and others helped push the movement forward. While Agile has gained wide adoption, it's also been misunderstood and criticized, often because of its cultural challenge to traditional corporate mindsets.

Ultimately, the Agile movement sought to bring **balance**, restore **credibility to methodologies**, and build **communities** of developers who value doing meaningful work in supportive environments. [\[1\]](#)

Key principles of agile methodology

1. **Customer Satisfaction through Continuous Delivery:** Agile places a high value on delivering working and potentially shippable increments of the product at the end of each iteration. This allows customers to see tangible progress regularly and provides opportunities for feedback.
2. **Embracing Change:** Agile recognizes that requirements can change throughout a project's lifecycle. It welcomes changes in requirements, even late in the development process, and provides mechanisms for adapting to evolving needs.
3. **Collaboration and Communication:** Agile emphasizes close collaboration among cross-functional teams, including developers, testers, and business representatives. Daily stand-up meetings and regular communication channels facilitate quick problem-solving and alignment of goals.
4. **Iterative and Incremental Development:** Agile projects are divided into small, manageable iterations or sprints, each typically lasting two to four weeks. At the end of each iteration, a potentially shippable product increment is delivered, allowing for continuous improvement and adjustment.
5. **Self-Organizing Teams:** Agile teams are encouraged to be self-organizing and cross-functional, with members collaborating to accomplish project goals. This autonomy fosters a sense of ownership and accountability among team members.

The Agile methodology is not a one-size-fits-all solution but can be adapted to various project types and industries. It provides a flexible and responsive framework that enables teams to deliver value quickly, adapt to changing requirements, and foster continuous improvement. [\[2\]](#)

II.3 Scrum history and basics

II.3.1 The origin of scrum

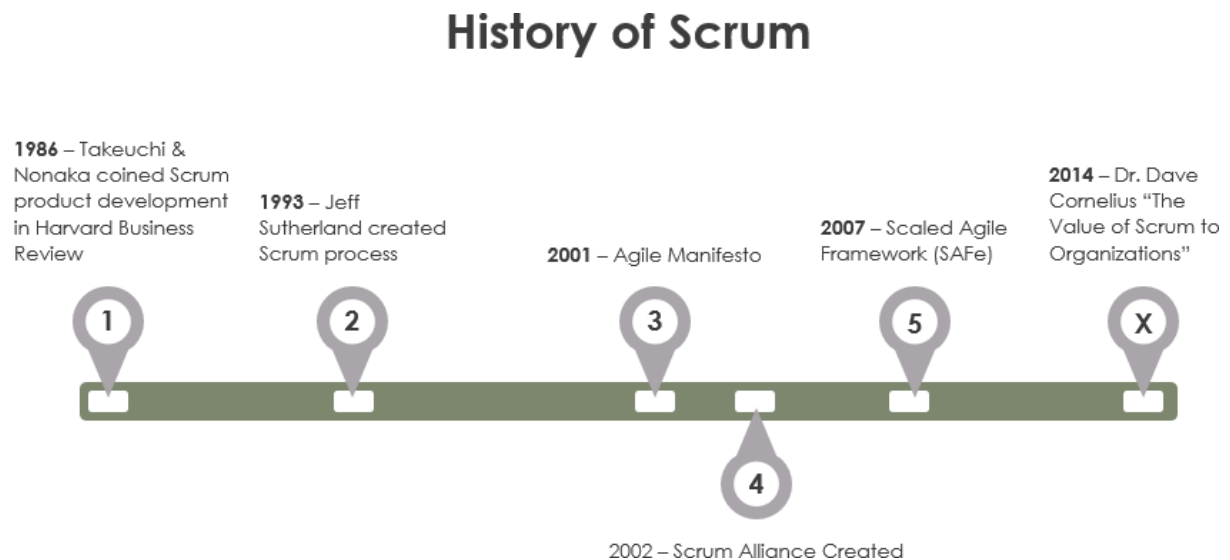


FIGURE 1 - History of scrum [\[9\]](#)

As shown in Figure 1, Agile thinking is deeply influenced by the best practices of Japanese industry, especially the lean principles promoted by Toyota and Honda, and the knowledge management strategies developed by Takeuchi Hiroyuki and Nogami Yujiro. Influenced by the above ideas and research on software projects worldwide, Jeff Sutherland first defined the Scrum process for the software development industry at Easel in 1993 and began to implement it.

1986 – Takeuchi & Nonaka coined Scrum product development in Harvard Business Review

1993 – Jeff Sutherland used Scrum for software development for the first time.

1995 – Jeff Sutherland and Ken Schwaber standardized the Scrum framework and made it publicly available on OOPSLA 95.

2001 – The Agile Manifesto and Principles were released, and the Agile Alliance was established. Scrum is one of the agile methods.

200 – Ken Schwaber and Mike Beedle launched the first Scrum book, Scrum Agile Software Development.

2002 – Ken Schwaber and Mike Cohn co-founded the Scrum Alliance. [\[3\]](#)

II.3.2 Scrum values

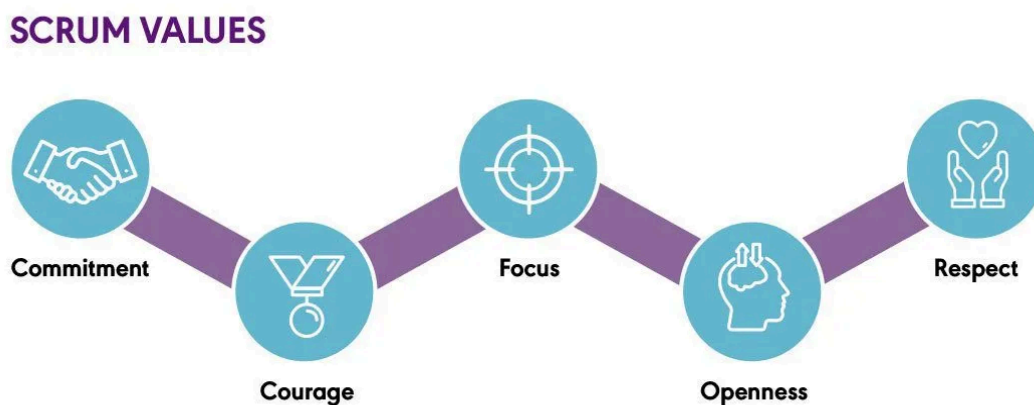


FIGURE 2 - Scrum values [\[10\]](#)

As shown in Figure 2, Commitment, Openness, Focus, Respect, and Courage are the five Scrum values. To achieve Scrum project management success, each Scrum team member must adhere to these values.

➤ Commitment

As the name implies, one of the key Scrum values essential in the success of any Scrum project management is commitment. All teams and team members must adhere to the Scrum principles and collaborate as a unit to achieve common goals.

➤ Focus

Focus is another Scrum model value that is important in ensuring that each sprint is completed on time and with the highest possible quality of work. To accomplish this objective, each team and Scrum team member must be more focused on their responsibilities and objectives.

➤ Openness

Transparency or openness among Scrum teams and team members is crucial for Scrum project management or for the Agile Scrum process. Each of the project's teams and stakeholders is expected to be open and honest about their progress and challenges in completing tasks.

➤ Respect

Each member of the Scrum team is expected to respect one another. When it comes to respect, an individual's role or position should not come into play. Respect for each individual will eventually result in a positive and healthy workplace where everyone will be able to give their best.

➤ Courage

Scrum team members should have the belief and confidence to take the right step and do the right thing, and work on problems that are complex in nature. This is only possible if they are open about their challenges and work, as well as transparent, focused, and committed to their work.

II.3.3 Scrum roles

SCRUM TEAM

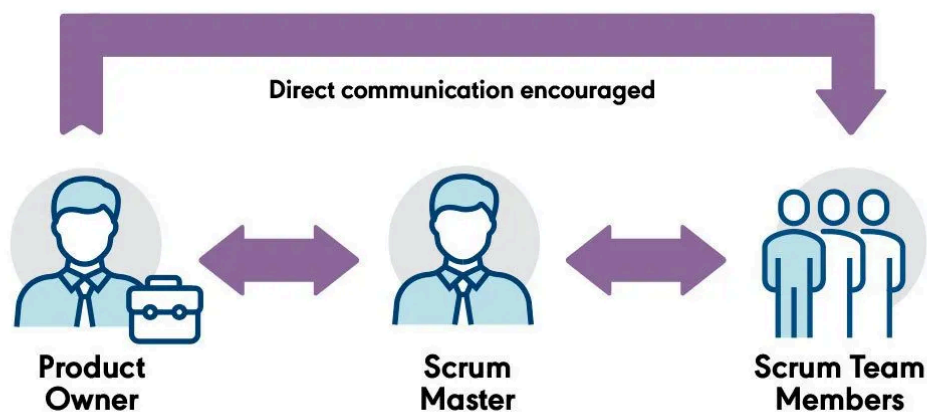


FIGURE 3 - Scrum team [10]

The Scrum team primarily consists of three roles: Product Owner, Scrum Master, and Scrum Team. Each of these roles has a specific set of responsibilities to fulfill.

➤ Product Owner

Product Owner, also known as PO, is an individual, who is in charge of ensuring that products are delivered on time and with the best quality. The Product Owner is also in charge of maximizing the product's value and the work of the development team.

The following are some of the primary duties of a Product Owner:

- Accountable for the product's success
- Product backlog management and prioritization
- Maximizing value to product quality
- Stakeholder management
- Direct the project in the right direction
- Defining and announcing releases

➤ Scrum Master

The Scrum Master is a position that is responsible for providing all guidance, resources, and support to the Scrum team in order to help them work at their best. It is their responsibility to create an environment for the Scrum team in which everyone can work without distraction.

Some of the primary responsibilities of a Scrum Master include:

- Keeping the team focused, organized, and motivated
- Facilitating Scrum events like daily stand-ups and meetings as needed
- Making sure that everyone in the team understands the scope and goal of the project
- Conducting a retrospective sprint
- Helping in grooming the product backlog

➤ Scrum Team

A Scrum team, also known as a development team, is a group of people who work together to complete and deliver the requested product, feature, or other requirements. The Scrum team prioritizes items from the product backlog based on the Scrum master's guidance.

Some of the Scrum team's key responsibilities

include:

- On-time completion of the sprint work
- In charge of managing the product backlog
- To maintain transparency throughout the sprint
- To inspect and adapt the work on a regular basis
- Use data to identify the best development practices
- Test prototypes and products

Each of these Scrum roles is critical to the successful implementation of the Agile Scrum process.

(All content adapted from [\[4\]](#)).

II.3.3 Scrum processes

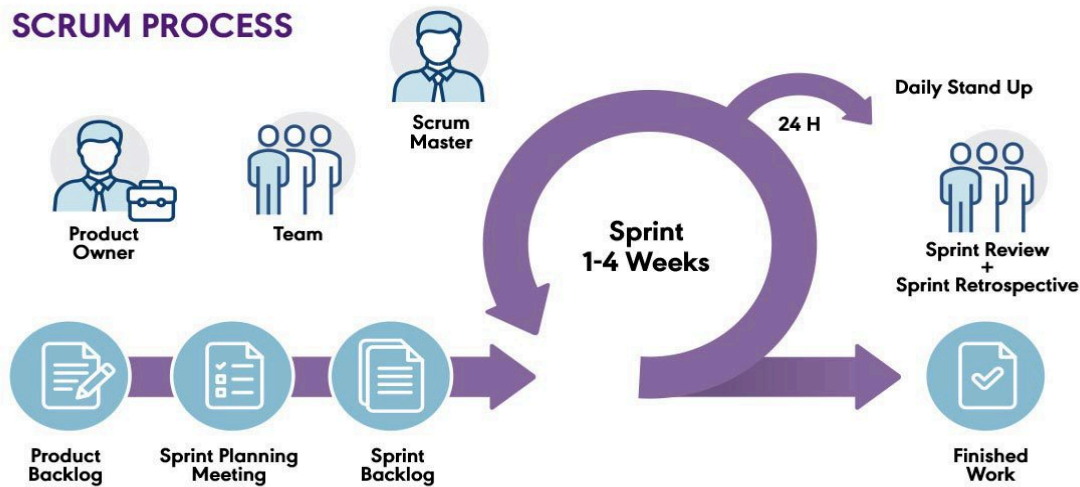


FIGURE 4 - Scrum process [10]

In essence, Scrum tools and techniques allow for large tasks to be broken down into manageable chunks that can be completed in short cycles – sprints – of one month or less.

The main aim of the Scrum process is to deliver value to the customer early and continuously throughout the project's lifecycle. To this end, those features and product increments that provide the most value for the customer are ordered so that they are developed and delivered first.

To facilitate this process, the Scrum framework relies on the Scrum team and its roles, artefacts, and ceremonies and typically uses a visual representation of the workflow to keep all team members in the loop on WIP and what's to come. These elements collectively enable the execution of projects with agility and effectiveness.

Scrum team – The Scrum team is a small self-organising, cross-functional team consisting of a Scrum master, a product owner, and developers. Between them, they are accountable for delivering an increment of valuable work during a sprint.

Artefacts – Artefacts refer to key information the team uses to define the product and work to be done, such as product backlog, sprint backlog, and product increment, and provide visibility into the development process and progress being made.

Ceremonies – Ceremonies are the prescribed events or meetings that ensure everyone is on the same page regarding goals, progress, and improvements. They include sprint planning, daily stand-ups, sprint reviews, and sprint retrospectives (see terminology below).

By promoting transparency, collaboration, and adaptability, these interrelated components help the Scrum team do their job. Together, they enable the team to plan, execute, and inspect their work, leading to continuous improvement and the delivery of high-quality products that meet stakeholders' needs.

(All content adapted from [\[4\]](#)).

II.4 Estimation in project management

II.4.1 Project estimation

Project estimation forecasts the resources, time, and costs required to complete a project. It involves analyzing available information, such as project scope, historical data, and industry benchmarks, alongside educated assumptions about potential risks and challenges.

This process aims to create a realistic picture of the project's overall effort and expenditure. Effective project estimation methods allow project managers to set achievable deadlines, allocate resources, and inform decision-making throughout the project life cycle. [\[6\]](#)

II.4.2 Importance of estimating a project

Project estimation is a crucial foundation for successful project execution. Accurate estimates empower project managers to make informed resource allocation decisions. By understanding the anticipated workload, you can assign the right people to the right tasks at the right time, preventing the overallocation or underutilization of resources. Realistic project estimates translate to workable schedules. Knowing the time necessary for each project phase helps managers establish achievable deadlines and milestones. This transparency fosters trust with stakeholders and mitigates the risk of missed deadlines or schedule overruns.

Project estimation plays a crucial role in managing stakeholder expectations. By communicating realistic timelines and resource requirements upfront, stakeholders can make informed decisions to avoid potential disappointment or frustration with project progress. [\[6\]](#)

II.4.3 Improving project planning through accurate estimation

Estimates require time to prepare. If asked for an on-the-spot estimate, reserve the right to revise it after consulting your team.

There are three main types of estimates:

- 1. Rough Order of Magnitude (ROM):**
 - Done at project initiation.
 - Accuracy ranges from -25% to +75%.

- Can be completed in hours or a day.
- 2. **Budget Estimate:**
 - More accurate, with a range of -10% to +25%.
 - Done during project planning.
 - Requires several days to weeks.
- 3. **Definitive Estimate:**
 - Most accurate, within -10% to +10%.
 - Requires detailed planning and task breakdowns.
 - Developed progressively over weeks or months.

it's important to explain them to stakeholders. Estimates should be based on proper analysis, understanding of tasks, and past experience, not random guesses. Always verify assumptions and include details with the person providing the estimate. [\[7\]](#)

II.4.4 Agile story points estimation

Story points are an estimation technique used in Agile project management methodologies to help your team scope the effort required to complete a task. [\[12\]](#)

Estimation, a form of understanding the size of the work, is a collaborative process in which the developers discuss the effort of completing an item from the product backlog. The question is, "If we were to implement this product backlog item fully, what is the work involved, and what is the effort of that work?"

As part of a discussion of the effort involved, the developers may factor in the following:

- The complexity of the work
- The amount of work
- Risk and uncertainty

While the variables above are the most traditional effort-affecting variables, they are not the only ones. As a unique team in a specific domain, there could be other factors that influence how you size a Product Backlog Item. [\[13\]](#)

II.5 Conclusion

Understanding Agile and Scrum methodologies, along with project estimation techniques, is essential for navigating the complexities of modern software development. These frameworks emphasize adaptability, collaboration, and continuous delivery, which are principles that support the rapid evolution of project requirements and user needs. Scrum, in particular, offers a structured yet flexible approach, enabling teams to work efficiently and deliver value incrementally.

Accurate project estimation complements these methodologies by providing a foundation for effective planning and resource management. It helps bridge the gap between vision and execution, ensuring that goals are both ambitious and achievable.

By grounding our proposed AI-powered Requirement Management Application in these foundational principles, we aim to address real-world challenges teams face, particularly the ongoing struggle to manage evolving requirements while maintaining project clarity and efficiency. This theoretical grounding not only frames the problem but also highlights the importance of intelligent tools that can support and enhance agile project environments.

III. Problem statement

III.1 Introduction.....	17
III.2 Challenges of incomplete or hidden requirements.....	17
III.3 Case studies.....	18
III.4 Analysis of delays due to hidden or incomplete requirements.....	18
III.5 Conclusion.....	19

III.1 Introduction

Requirement management forms the critical foundation of successful software development, yet it remains one of the most persistent challenges facing modern engineering teams. As products grow increasingly complex and stakeholder expectations escalate, the process of capturing, documenting, analyzing, and tracking requirements has become a formidable obstacle course that can make or break project outcomes. Research indicates that a staggering 71% of software projects fail due to poor requirements management. [8]

III.2 Challenges of incomplete or hidden requirements

A study [11] classified the top 10 problems in requirements engineering across 228 companies from 10 different countries. We selected the top 3 problems from them.

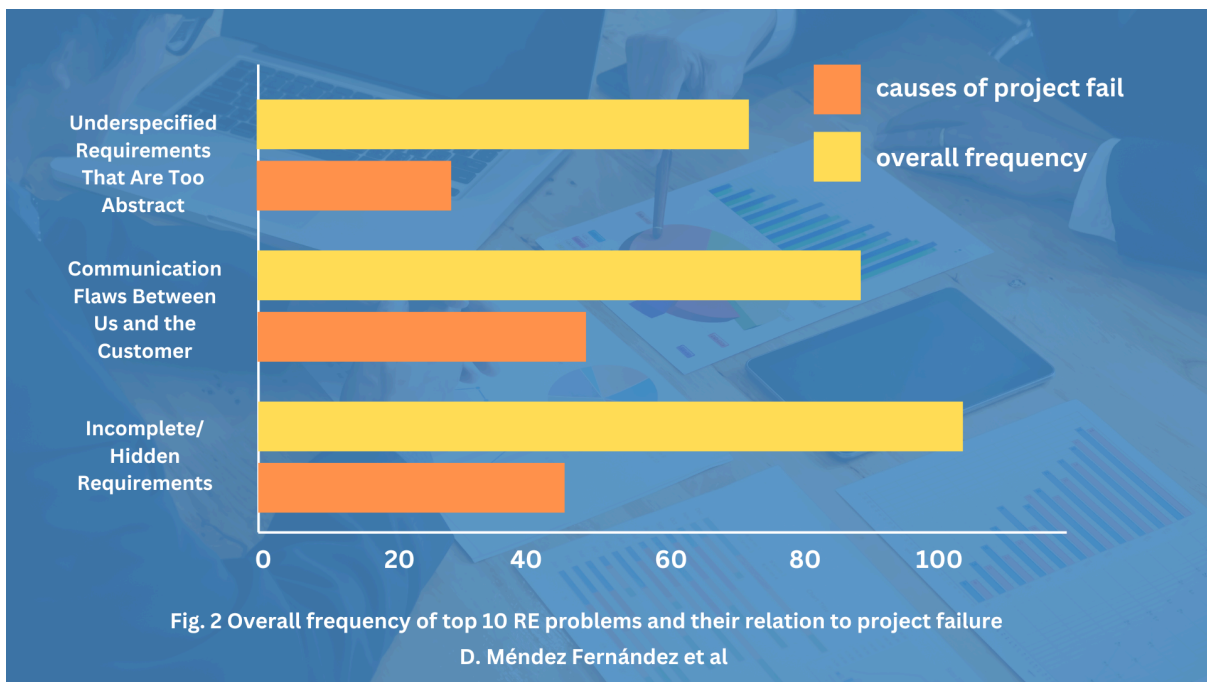


FIGURE 5 -Naming the pain in requirements engineering contemporary problems, causes, and effects in practice [11]

As shown in figure 5, We noticed that incomplete or hidden requirements had an overall frequency of more than 100, making it the top issue on the list. It also caused more than 40 project failures after communication.

We can improve communication by organizing more meetings with the customer. However, fixing incomplete or hidden requirements requires much more effort.

Additionally, we observed that underspecified requirements are a major cause of incomplete and hidden requirements.

III.3 Case studies

Throughout our experience at **Five Angles** and **StackInTech**, several projects were significantly impacted by hidden or incomplete requirements. Below, we present examples of projects from both companies that faced notable delays, along with an analysis of the main contributing factors.

Project	Planned Duration	Actual Duration	Delay (Months)	Primary Cause
Education Platform	2 months	3 months	+1	Missing content management details
Support Community Platform	2 months	5 months	+3	Undefined user interactions
Short Video Platform	2 months	6 months	+4	Performance + underspecified functions

III.4 Analysis of delays due to hidden or incomplete requirements

Across all these projects, a common pattern was observed:

- **Initial requirement sessions were insufficient** in uncovering real user needs.
- **Evolving client expectations** without structured change management processes introduced scope creep.
- **Underspecified technical details** created uncertainties during development.

These factors led to extended timelines, increased costs, and rework, reinforcing the need for better requirements engineering practices.

III.5 Conclusion

The recurring challenges faced by Five Angles and StackInTech, as well as the broader industry data, clearly highlight a critical gap in current requirements engineering practices. Incomplete, hidden, and underspecified requirements continue to cause significant project delays, increased costs, and compromised project outcomes. While improving communication with customers can help, it alone is not enough to eliminate these issues.

To effectively address these challenges, there is an urgent need for a systematic, automated approach to requirement management – one that enhances early detection of gaps, clarifies ambiguities, and adapts dynamically to evolving stakeholder needs. By leveraging advanced technologies such as artificial intelligence, natural language processing, and intelligent requirement validation tools, engineering teams can significantly improve the quality of requirements, reduce risks, and set projects on a clearer path to success.

IV. Proposed solution: implementation

IV.1 Introduction.....	21
IV.2 Project conception.....	21
IV.2.1 Use-case diagram.....	22
IV.2.2 Sequence diagrams.....	26
IV.2.2.1 Automatic generation of user stories based on epic and project context.....	26
IV.2.2.2 AI-Assisted developer consultation to refine user stories.....	28
IV.2.2.3 Automatic estimation based on 3 historical user stories.....	30
IV.2.3 Entity-relationship diagram.....	32
IV.2.3.1 Users and roles (Section 1).....	32
IV.2.3.2 Domain, platform and technologie (DPT) (Section 2).....	32
IV.2.3.3 Epics and user stories (Section 3).....	33
IV.2.3.4 Tasks (Section 4).....	33
IV.3 Detailed explanation.....	34
IV.3.1 Users and roles.....	34
IV.3.2 Domain, platform and technologie.....	35
IV.3.3 Epic and user story.....	36
IV.3.4 Tasks and acceptance criteria.....	37
IV.4 Tools and technologies used.....	38
IV.5 Project visual overview.....	40

IV.1 Introduction

The recurring challenges in requirements engineering , including hidden, incomplete requirements, call for a transformative solution. To address these issues, we propose the development of an **AI-powered Requirement Management Application**.

This web-based application aims to **augment and automate** key phases of the requirements engineering process. By leveraging **artificial intelligence (AI)**, the tool assists teams in creating high-quality, well-defined requirements faster and more efficiently.

The system will accept **epics and basic descriptions** as input, and will then automatically generate **user stories, acceptance criteria, and development tasks**, customized to the context of the project. Product Owners (POs) and stakeholders will simply **verify, and validate** the AI's suggestions, rather than drafting detailed requirements manually from scratch.

Additionally, the application will offer advanced capabilities such as **user flow generation, AI-assisted developer consultation, and automatic effort estimation**. Once the requirements are finalized, they can be **seamlessly exported to Jira**, ensuring a smooth transition into the development phase.

This solution is designed to **minimize project risks, reduce delays, and improve the overall quality and success rate** of software projects.

IV.2 Project conception

The conception of the AI-powered Requirement Management Application stems from real-world challenges experienced by organizations like **Five Angles** and **StackInTech**, as well as broader industry observations. Repeated project failures, scope creep, missed deadlines, and increased costs are often traced back to **poorly defined requirements** at the earliest stages of project development.

Traditional approaches rely heavily on manual processes, subjective interpretation, and extensive back-and-forth communication, leading to a high risk of human error and misalignment with stakeholder expectations.

Recognizing these gaps, the project was conceived with the following core ideas:

- **Start from simplicity:** Instead of overwhelming users with complex templates and detailed forms, allow them to input simple **epics and brief descriptions**.
- **Leverage AI for smart expansion:** Use AI to automatically **expand** simple inputs into fully detailed **user stories**, each equipped with **acceptance criteria** and **task**

breakdowns.

- **Focus on validation, not creation:** Shift the role of Product Owners and stakeholders from manually creating requirements to **verifying, and validating** AI-generated outputs, significantly reducing time and cognitive load.
- **Continuous refinement with AI assistance:** Allow users to ask AI for **additional clarifications, suggest refinements, generate user flows, and estimate efforts**, promoting continuous improvement of requirements throughout the project lifecycle.
- **Integrate seamlessly with development tools:** Ensure the generated outputs can be **exported directly to Jira**, aligning requirement engineering with agile development workflows.

The goal is not to replace human expertise, but to **augment and empower** it with AI-driven insights, creating a smarter, faster, and more resilient requirements management process.

IV.2.1 Use-case diagram

The use-case diagram illustrates the interaction between the AI-Powered Requirement Management Application and its primary actor.

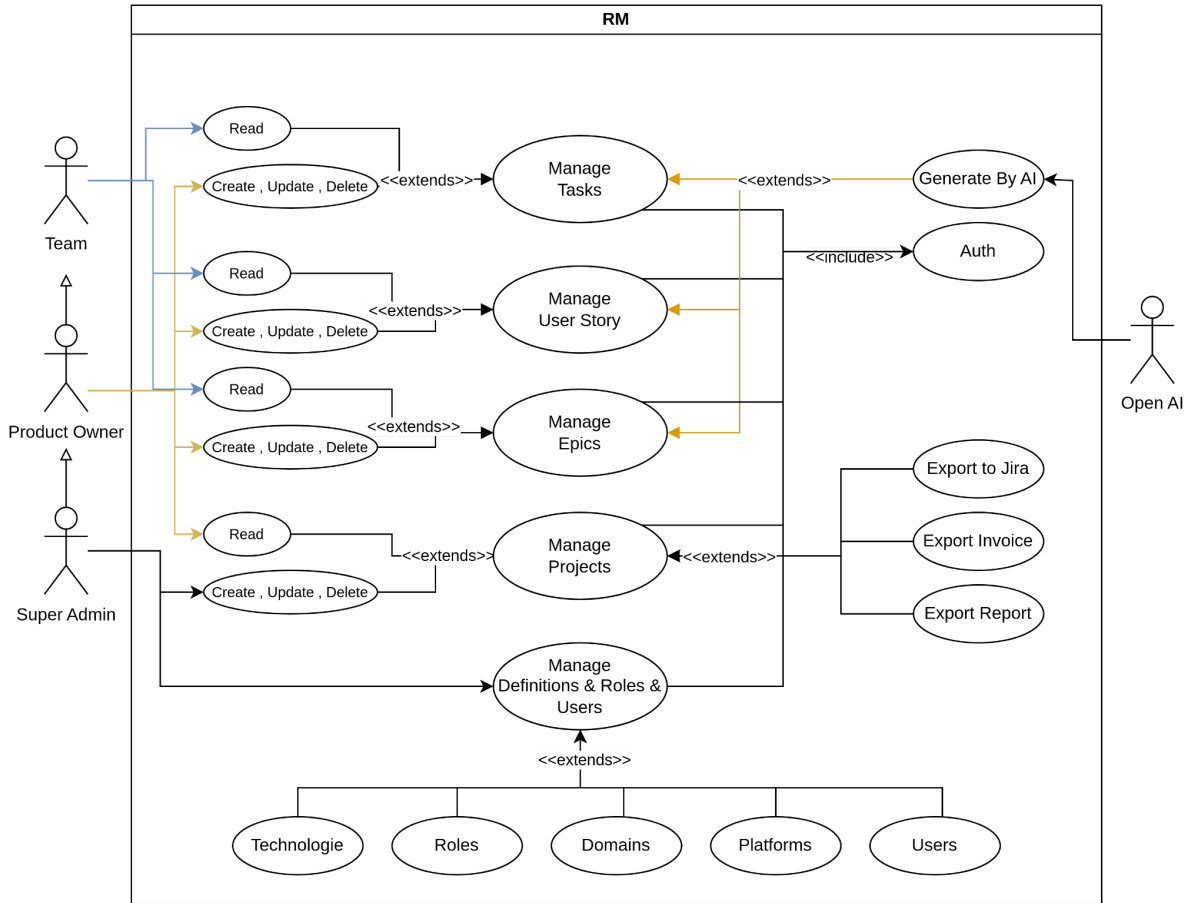


FIGURE 7 - Use-Case Diagram

As shown in figure 7, this use-case diagram illustrates the core functionalities and interactions within the requirements management system (RM). It identifies the system actors and how they interact with various use cases to manage project-related entities and configurations.

Use Case	Description
Manage Projects	Creates, updates, deletes, and reads projects to organize project work.
Manage Epic	Creates, updates, deletes, and reads epics, which are large project features.
Manage User Story	Creates, updates, deletes, and reads user stories to capture project requirements.
Manage tasks	Creating, updating, deleting, and reading tasks for project work.
Manage definitions & roles	Sets up technologies, roles, domains, platforms, and users in the system.
Read create update delete	Performs basic create, update, delete, and read actions on tasks, stories, epics, or projects; some roles may have read-only access.
Auth	Ensures users are authenticated before access to the application.
Generate by AI	Uses AI to automatically create or assist with epics, user stories, and tasks.
Export to jira	Exports user stories to Jira for tracking in another tool.
Export invoice	Exports invoices based on projects for billing.
Export report	Exports reports based on project progress.
Roles	Defines user roles and attaches permissions in the system.
Technologies	Manages the tech tools or stacks used in projects.
Domains	Manages domains used in projects.
Platforms	Manages platforms used in projects.
Users	Manages user accounts and access.

Actors:

- **Super admin:** A high-level administrator with full system access, responsible for configuring the system, managing projects, defining roles, and setting up custom permissions.

- **Product owner:** The individual who defines the product vision, prioritizes epics, and ensures the project aligns with business goals by managing high-level requirements.
- **Team:** A group of individuals whose role can be Developer, Designer, or QA (Quality Assurance), responsible for viewing tasks and user stories, epics to stay informed about project progress and requirements.

IV.2.2 Sequence diagrams

IV.2.2.1 Automatic generation of user stories based on epic and project context

Overview:

As shown in figure 8, this sequence diagram illustrates how the system leverages AI to generate relevant user stories by analyzing the associated epic and project metadata. It ensures initial requirements are quickly populated and aligned with the project's goals.

Workflow:

When the user initiates a request to generate user stories, the system begins by sending a notification indicating that the operation has started.

The system then retrieves the necessary project and epic metadata from the database. Once the metadata is fetched, it prepares a tailored prompt using this information and sends it to the OpenAI model.

Upon receiving a response from the AI, the system attempts to clean and format the data, keeping only the CSV content that represents the generated user stories. If successful, it proceeds to remove any existing draft user stories from the database, then inserts the new user stories in their place.

If all operations complete without error, the system sends a notification to the user confirming that the generation process was successfully completed.

However, if an error occurs at any point, whether during prompt preparation, AI response, data cleanup, or database operations, the system logs the issue and notifies the user that the generation process has failed.

This flow ensures that user stories are generated in a clean, automated manner and stored reliably, using AI assistance and database synchronization.

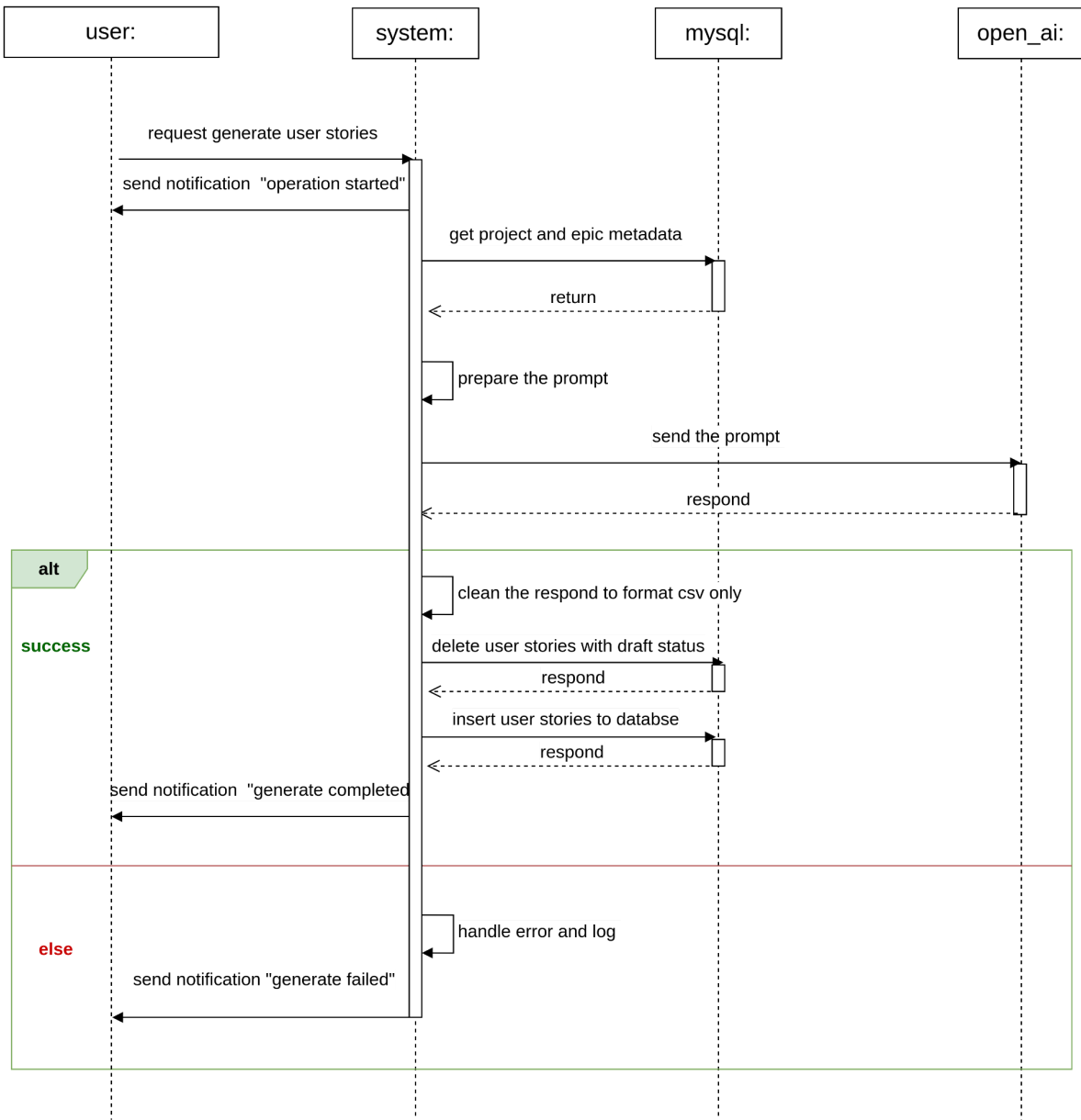


FIGURE 8 - Automatic generation of user stories based on epic and project context

IV.2.2.2 AI-Assisted developer consultation to refine user stories

Overview:

As shown in figure 9, this scenario shows how a developer interacts with the AI module to improve the clarity and completeness of a user story. The system provides suggestions or questions that help ensure all functional and non-functional aspects are considered.

Workflow:

When the user initiates a request to generate an AI-Assisted Developer description, the system first sends a notification indicating that the operation has started.

The system then retrieves the necessary user story description from the database. Once the relevant content is fetched, it constructs a prompt tailored for the AI and sends it to the OpenAI model.

Upon receiving a response from the AI, the system attempts to update the existing record in the database with the new AI-generated description.

If this process completes successfully, the system notifies the user that the generation has been completed.

In the event of any failure during the AI call, data preparation, or database update, the system handles the error internally, logs the issue, and notifies the user that the generation process has failed.

This workflow ensures that AI-generated developer descriptions are integrated seamlessly into the system while keeping the user informed of the operation's outcome.

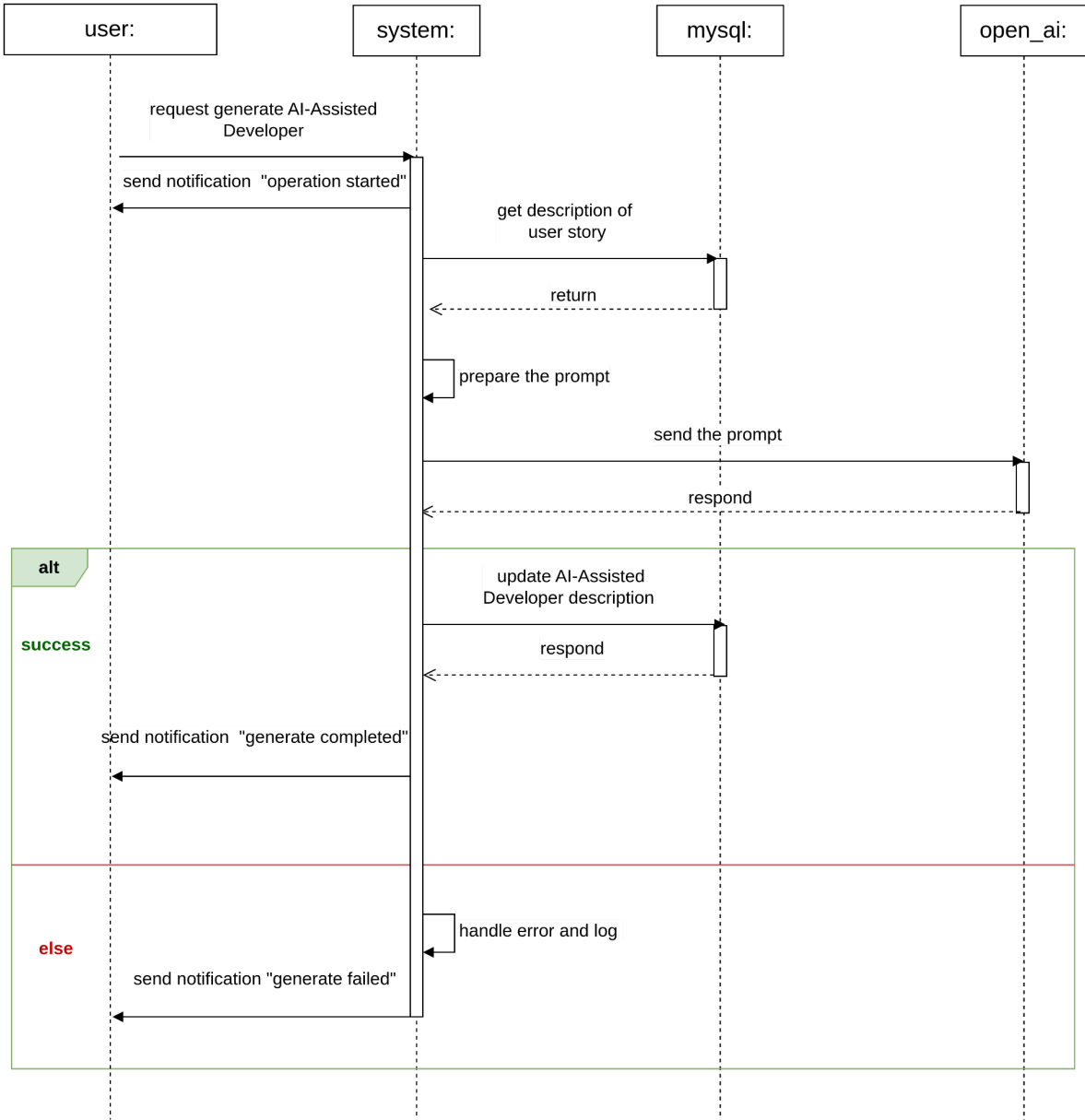


FIGURE 9 - AI-Assisted developer consultation to refine user stories

IV.2.2.3 Automatic estimation based on 3 historical user stories

Overview:

As shown in figure 10, once a user story is finalized, this diagram demonstrates the automated generation of story point estimations. We provide the AI with three user stories, each with a different story point value. Based on the description and acceptance criteria, the AI classifies the new user story into one of the existing estimation categories.

Workflow:

When the user initiates a request for AI-based estimation, the system begins by sending a notification that the operation has started.

The system then attempts to retrieve three completed user stories from the database, ensuring that each has a different story point value. These diverse examples are essential to provide context and accuracy for the estimation process.

If suitable user stories are found, the system constructs a prompt incorporating them and sends it to the OpenAI model for processing.

However, if no qualifying user stories are available, the system notifies the user that estimation cannot be performed due to insufficient data.

Once a response is received from the AI, the system tries to update the AI-Estimation description with the generated result.

If the operation is successful, the user receives a notification indicating the estimation has been completed.

In the event of an error during any phase, whether due to data retrieval, AI processing, or updating the database, the system logs the issue and notifies the user that the generation has failed.

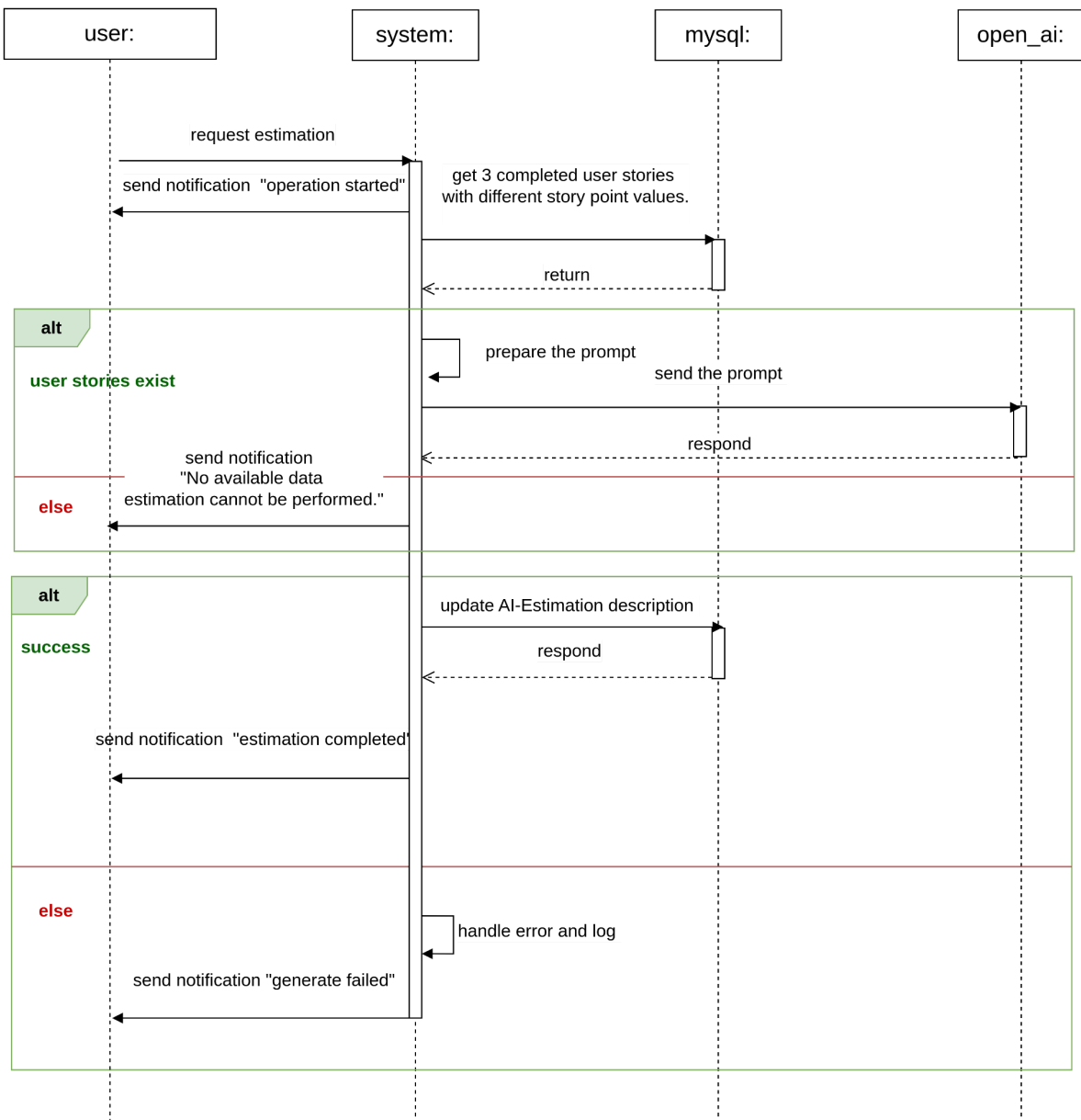


FIGURE 10 - Automatic estimation based on 3 historical user stories

IV.2.3 Entity-relationship diagram

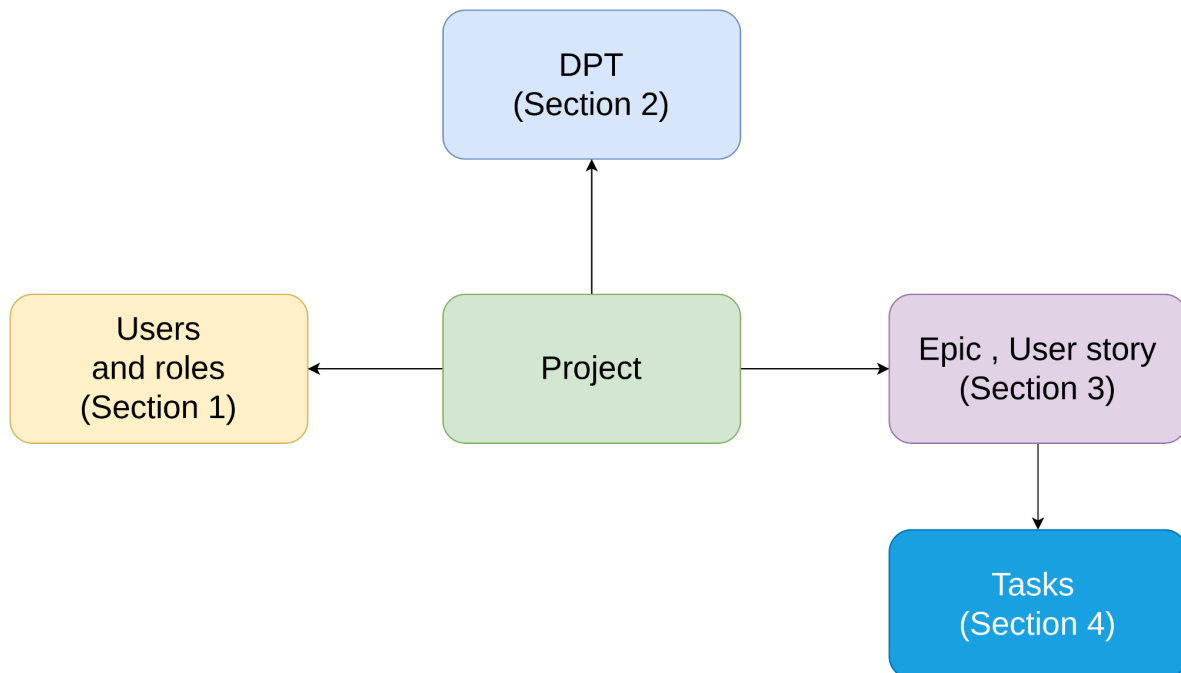


FIGURE 11 - high-level entity-relationship diagram

To manage the complexity of our large Entity-relationship diagram (ERD), we created a high-level overview diagram that breaks down the system into logical sections as shown in Figure 11.

This abstraction helps present the system architecture more clearly and allows for focused explanation of each component. The diagram is divided into four main sections, each representing a key part of the system:

IV.2.3.1 Users and roles (Section 1)

This section defines the users of the system and their associated roles and permissions. It outlines how different user types interact with the platform and what access levels they possess.

IV.2.3.2 Domain, platform and technologie (DPT) (Section 2)

This section organizes projects by domain, platform, and technology for structured categorization and resource allocation, defining project attributes and their technical/business contexts.

IV.2.3.3 Epics and user stories (Section 3)

Here, we break down the project requirements into epics and user stories. This section is crucial for understanding the scope of work and how features are structured from a product perspective.

IV.2.3.4 Tasks (Section 4)

This part details the actionable tasks derived from user stories. It is used to track development progress and ensure quality delivery.

Each section is explained in detail, outlining the corresponding entities and their roles. The central Project entity serves as the hub, connecting all parts and ensuring system cohesion.

IV.3 Detailed explanation

IV.3.1 Users and roles

As shown in figure 12, this section manages who can access the system, what they can do, and in which project context. It defines the different types of users, their roles within specific projects, and the permissions tied to each role.

- Users represent individuals using the system.
- Each user can be linked to multiple projects, with a specific role per project.
- Roles define the responsibilities or capabilities users have.
- Permissions are specific actions or access rights, grouped under roles.
- The system uses a role-based access control model, ensuring users have the appropriate permissions depending on their role in each project.

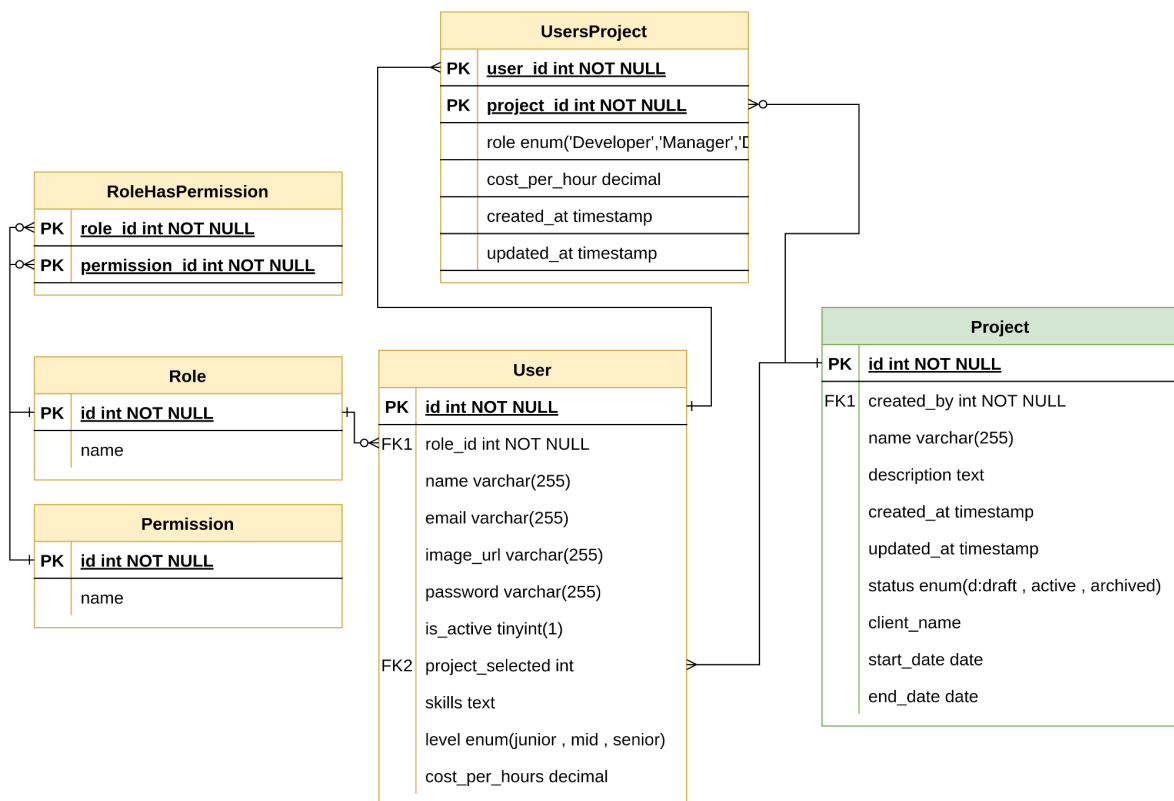


FIGURE 12 - Users and roles

IV.3.2 Domain, platform and technologie

As shown in figure 13, this section organizes projects by their domains, platforms, and technologies, enabling structured categorization and resource allocation. It defines the key attributes of projects and how they relate to broader technical and business contexts.

- Domain represents the business or application area a project belongs to (e.g., Healthcare, Finance).
- Platform indicates the runtime environment or framework used (e.g., Web, Mobile, Cloud).
- Technology refers to the specific tools, languages, or systems employed (e.g., React, Python, AWS).

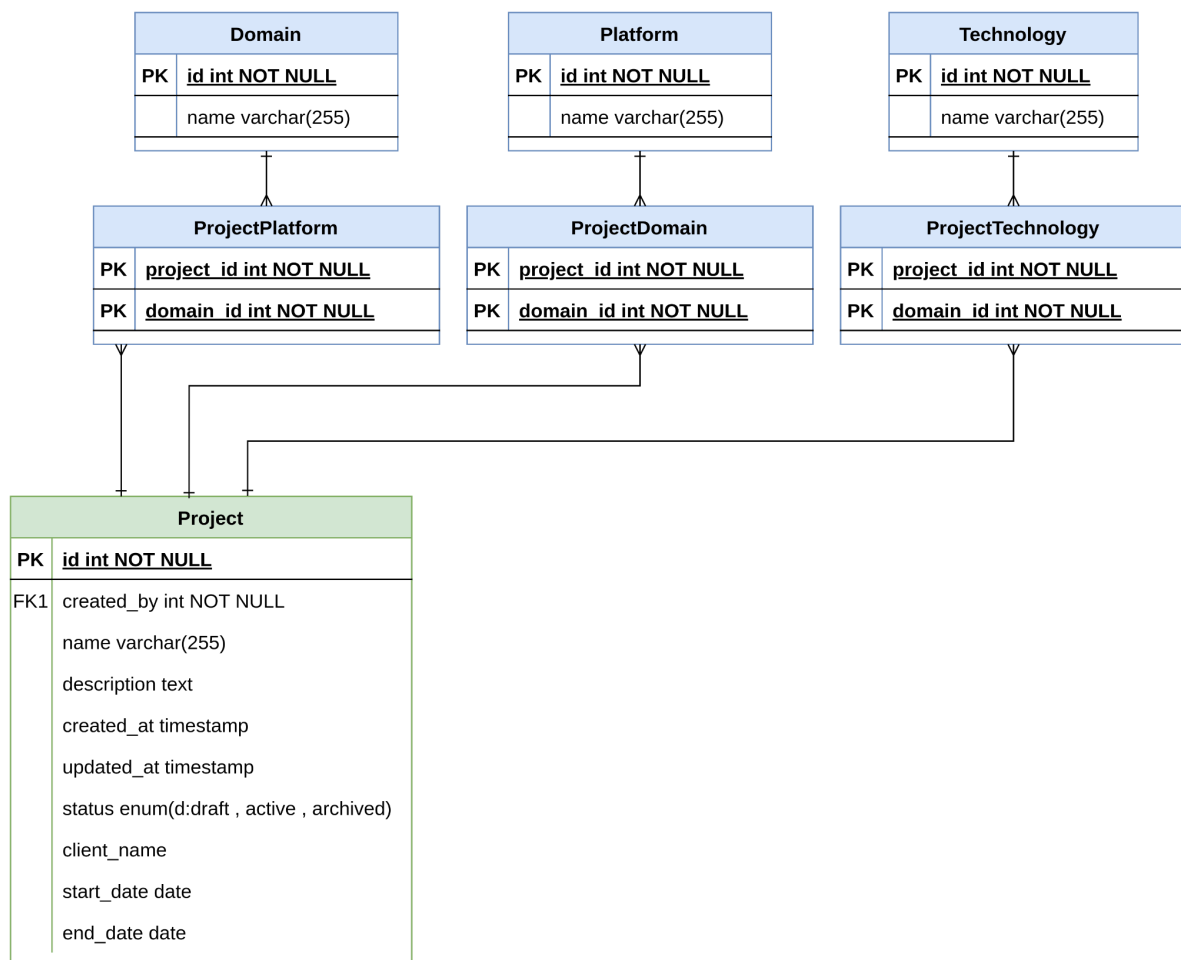


FIGURE 13 - Domain, platform and technologie

IV.3.3 Epic and user story

As shown in figure 14, this section outlines the hierarchical structure of project management, detailing the relationship between projects, epics, and user stories for effective task organization and tracking.

- Project represents the overarching initiative or goal that encompasses multiple related efforts.
- Epic denotes a large body of work that can be broken down into smaller user stories.
- User Story describes a specific, actionable requirement or feature from the end-user's perspective, forming the smallest unit of work.

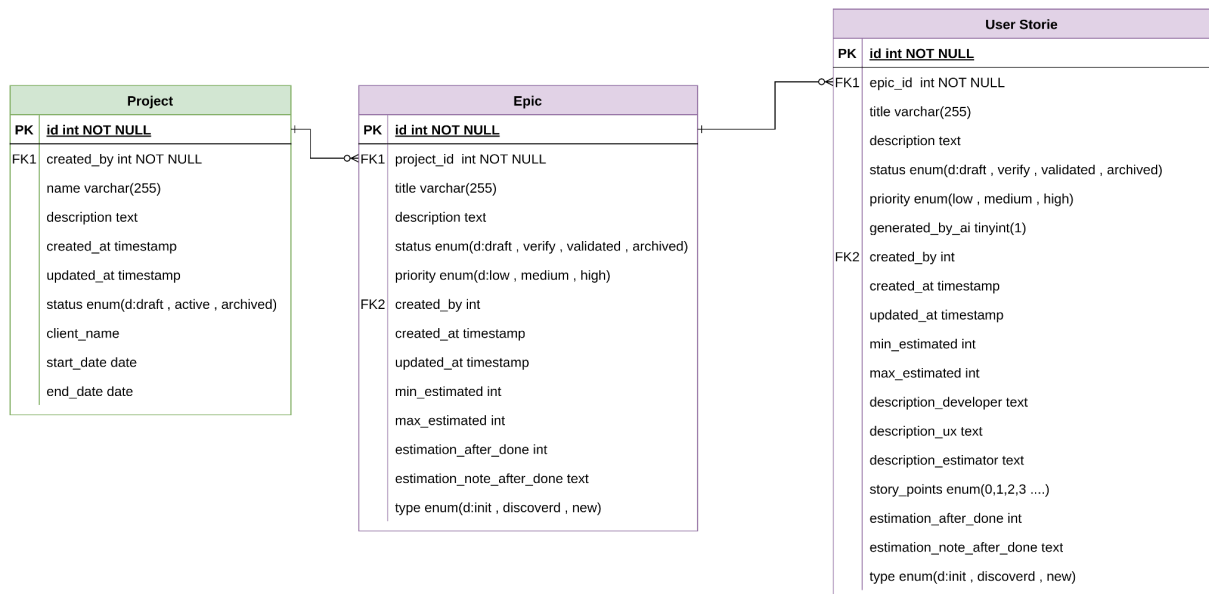


FIGURE 14 - Epic and user story

IV.3.4 Tasks and acceptance criteria

As shown in figure 15, this section illustrates the decomposition of a user story into its detailed components, ensuring clarity in requirements and execution.

- User Story defines a specific feature or requirement from the end-user's perspective, acting as the primary unit of work.
- Task represents the individual, actionable steps required to fulfill the user story.

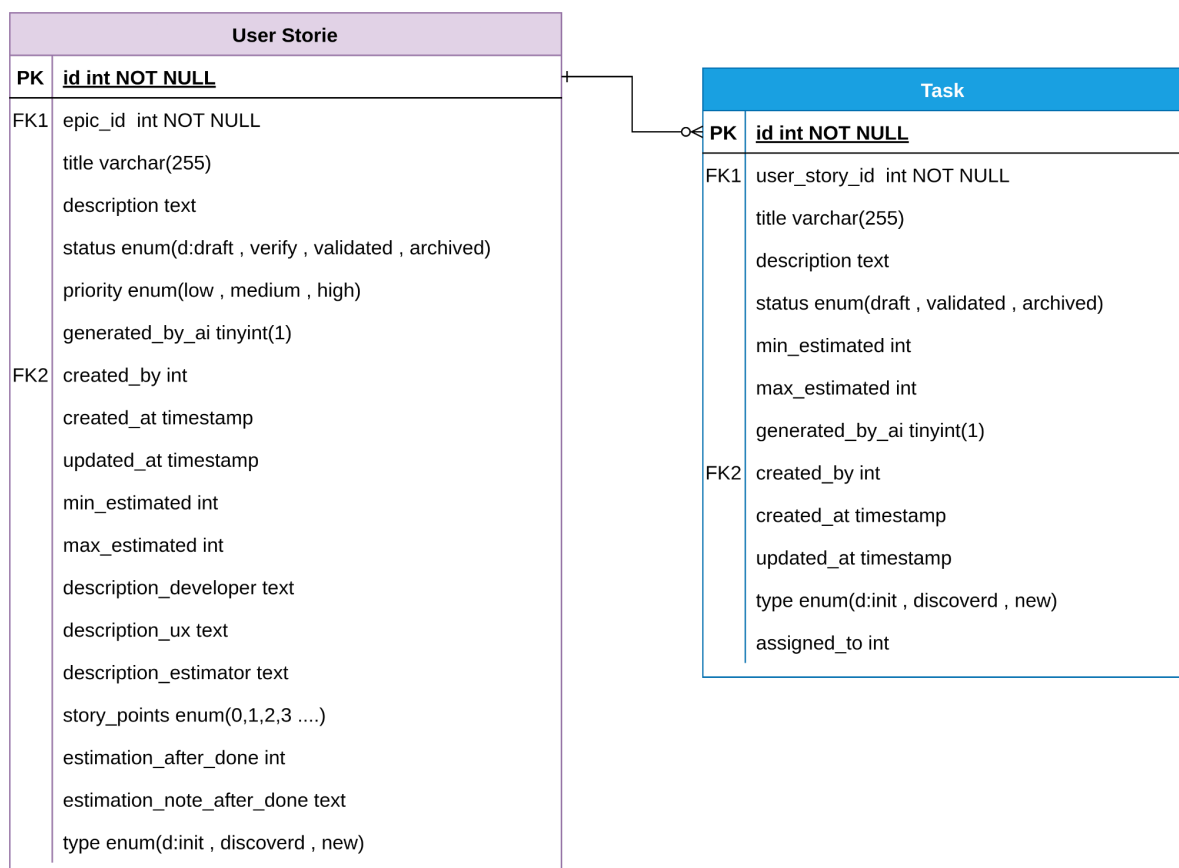


FIGURE 15 - Tasks

IV.4 Tools and technologies used

This section provides an overview of the tools and technologies employed in the project, focusing on their roles in development, deployment, and management. These tools collectively enable efficient coding, testing, collaboration, and hosting, forming the foundation for a robust and scalable application ecosystem.



FIGURE 16 - Tools and technologies used

Laravel

A PHP web application framework with expressive, elegant syntax. Laravel simplifies development by offering tools for routing, authentication, and caching. Think of it as a well-organized toolbox for building full-featured web applications. [\[25\]](#)

Docker

A platform for developing, shipping, and running applications inside containers. Containers encapsulate software along with its dependencies, making it portable and consistent across environments. Imagine moving your whole development setup in a neatly sealed box. [\[14\]](#)

OVH

A global cloud infrastructure provider offering VPS, dedicated servers, and scalable cloud hosting solutions. Think of it as a data center you can rent to deploy your applications reliably. [\[15\]](#)

Filament

A modern admin panel, form builder, and table manager for Laravel applications. It lets

developers quickly build interactive dashboards and content management interfaces. Think of it as a sleek control panel for your Laravel app. [\[16\]](#)

React

A JavaScript library for building user interfaces, especially single-page applications. React enables developers to create interactive web pages by composing the UI from reusable components. Think of React as LEGO blocks for your website. [\[17\]](#)

Git

A distributed version control system that allows developers to track changes in their source code, revert to previous stages, and collaborate on projects. It acts like the memory of your project, recording every change. [\[18\]](#)

PHP

A widely-used open-source scripting language suited for web development. PHP is server-side and powers platforms like WordPress and Laravel. Think of PHP as the engine behind your website's dynamic behavior. [\[19\]](#)

GitHub

A cloud-based Git repository hosting platform. It facilitates collaboration, code reviews, and project management, combining version control with social coding. Think of it as Git with a team hub and social layer. [\[20\]](#)

MySQL

An open-source relational database management system used to store and manage structured data. It's especially popular in combination with PHP. Think of it as an organized digital filing cabinet for your app's data. [\[21\]](#)

Jira

A project management tool developed by Atlassian, widely used in agile software development for tracking issues, planning sprints, and managing tasks. Think of Jira as your team's digital project whiteboard. [\[22\]](#)

VSCoDe (Visual Studio Code)

A free, lightweight code editor developed by Microsoft that supports debugging, syntax highlighting, and extensions. Think of it as a smart notebook made specifically for programmers. [\[23\]](#)

OpenAI

An AI research and deployment company offering APIs like ChatGPT for natural language processing tasks. Developers integrate OpenAI to create intelligent chatbots, content generators, and AI-powered apps. Think of it as adding an AI brain to your software. [\[24\]](#)

IV.5 Project visual overview

This section provides a glimpse into the project's interface through selected screenshots, showcasing its design, layout, and user interaction elements to illustrate the overall user experience and functionality.

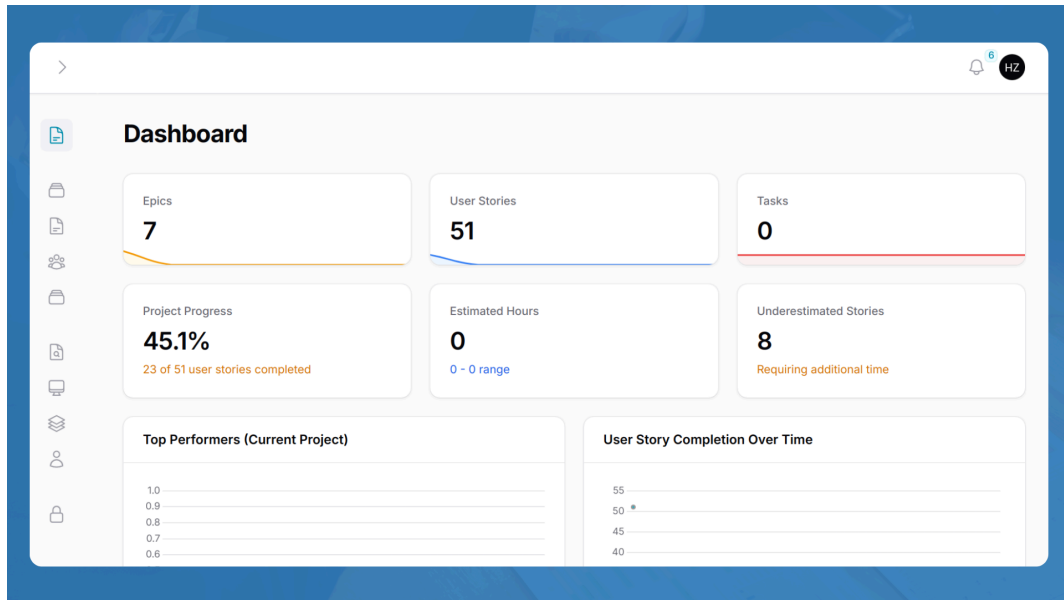


FIGURE 17 - Dashboard

Figure 17 presents the main dashboard screen, which provides users with an overview of key metrics. The dashboard typically includes summary statistics

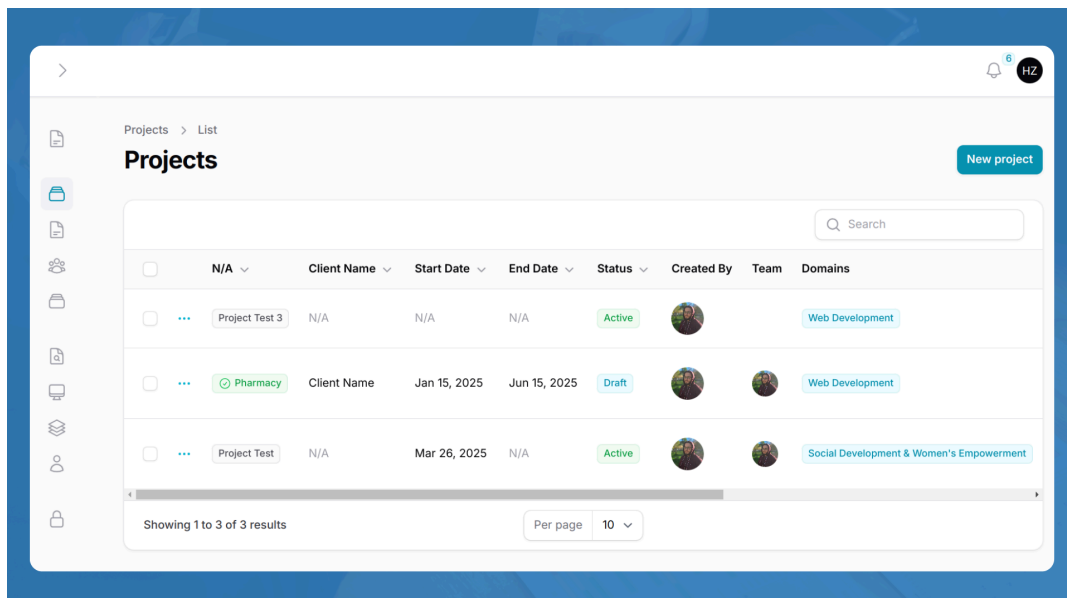


FIGURE 18 - Projects

Figure 18 shows the project management screen, where users can create, edit, and monitor projects. This interface allows for organizing project details.

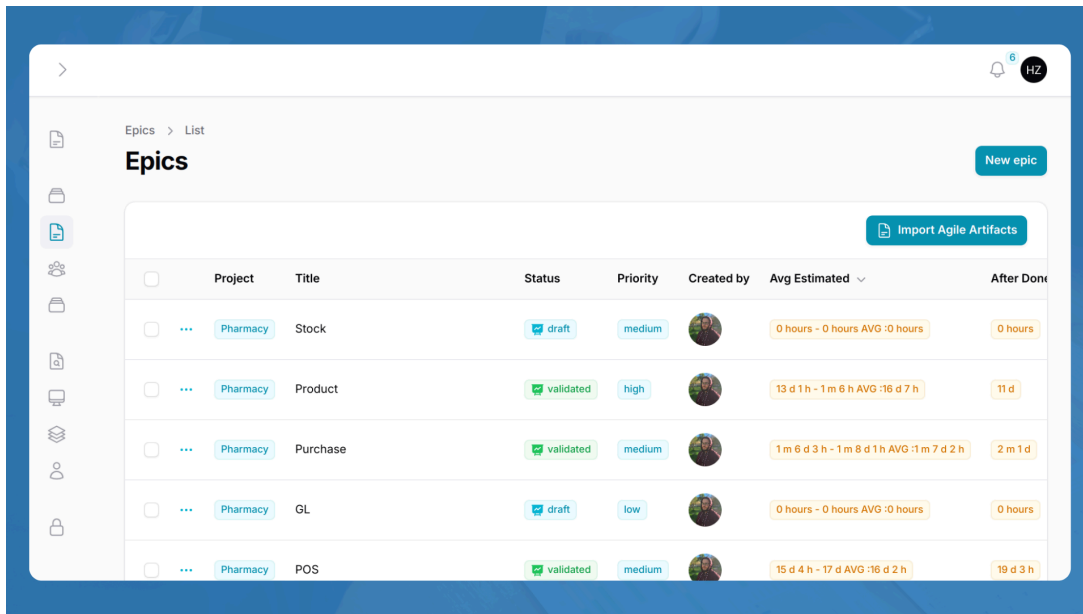


FIGURE 19 - Epics

Figure 19 illustrates the epic management screen, which enables users to define, organize, and oversee high-level features or objectives within a project. This screen provides tools to break down epics into user stories

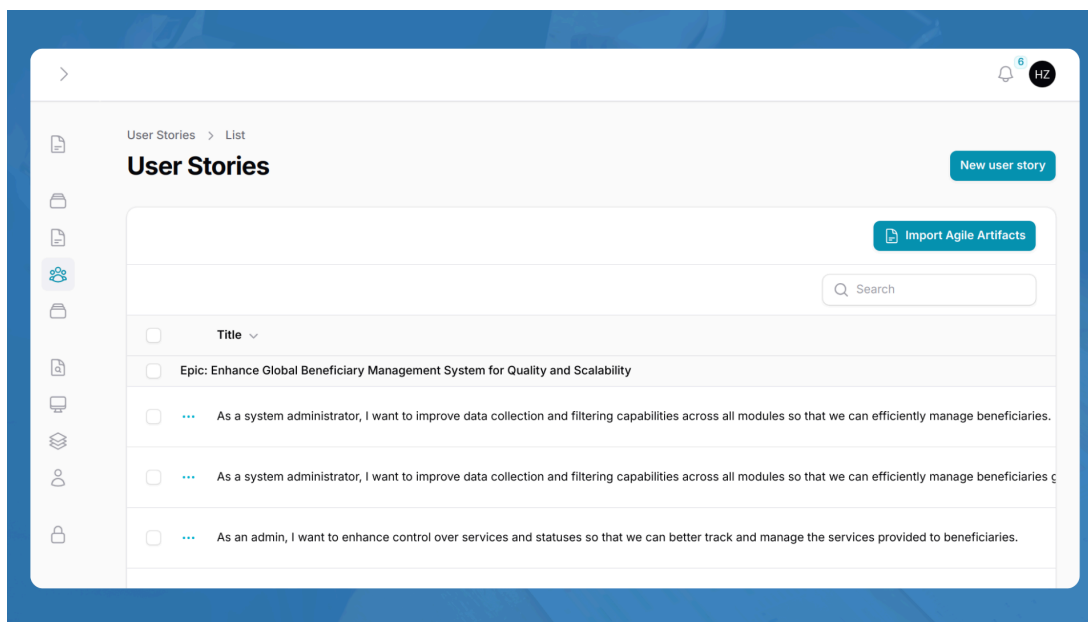


FIGURE 20 - User Stories

Figure 20 displays the user story screen, which allows users to create and manage user stories that describe specific functionalities or features from the end-user's perspective. This screen supports detailing requirements, setting priorities, and linking stories to related tasks or epics.

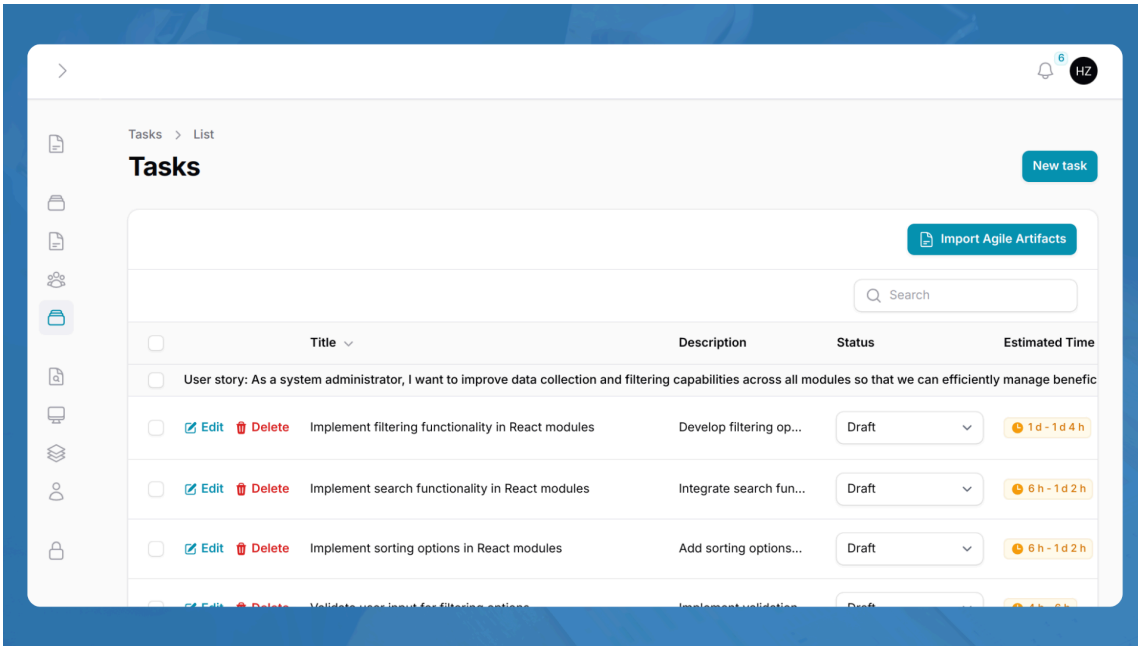


FIGURE 21 - Tasks

Figure 21 presents the task screen, where individual tasks related to user stories can be managed.

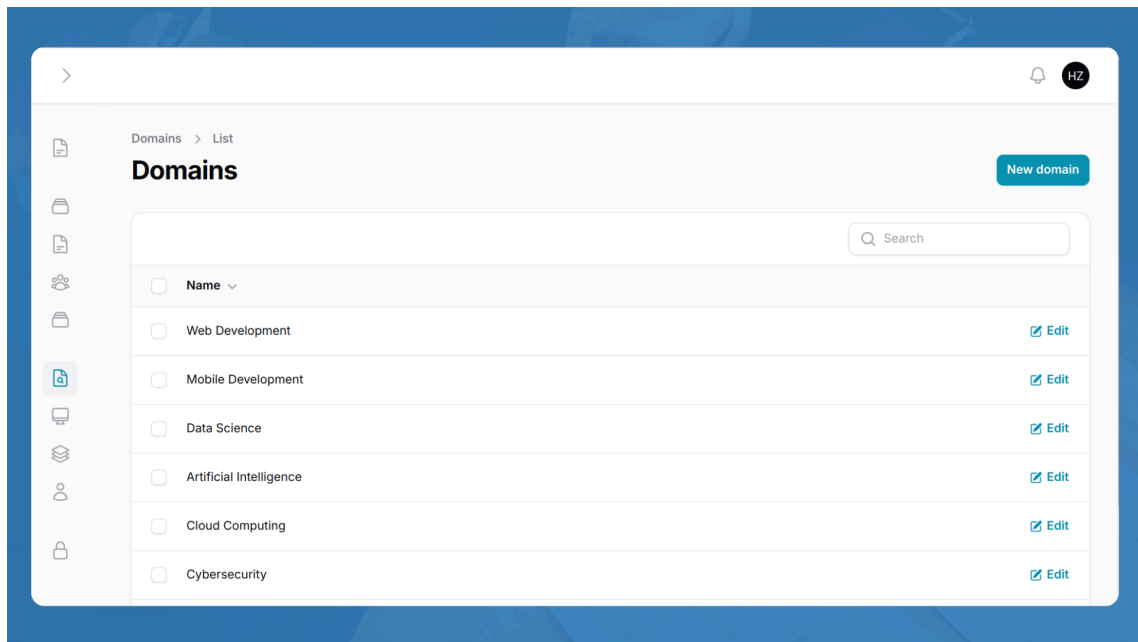


FIGURE 23 - Domains

Figure 23 presents the domains management screen, which enables users to define and manage domains within the system. A domain represents a conceptual or functional area that can be linked to one or multiple projects.

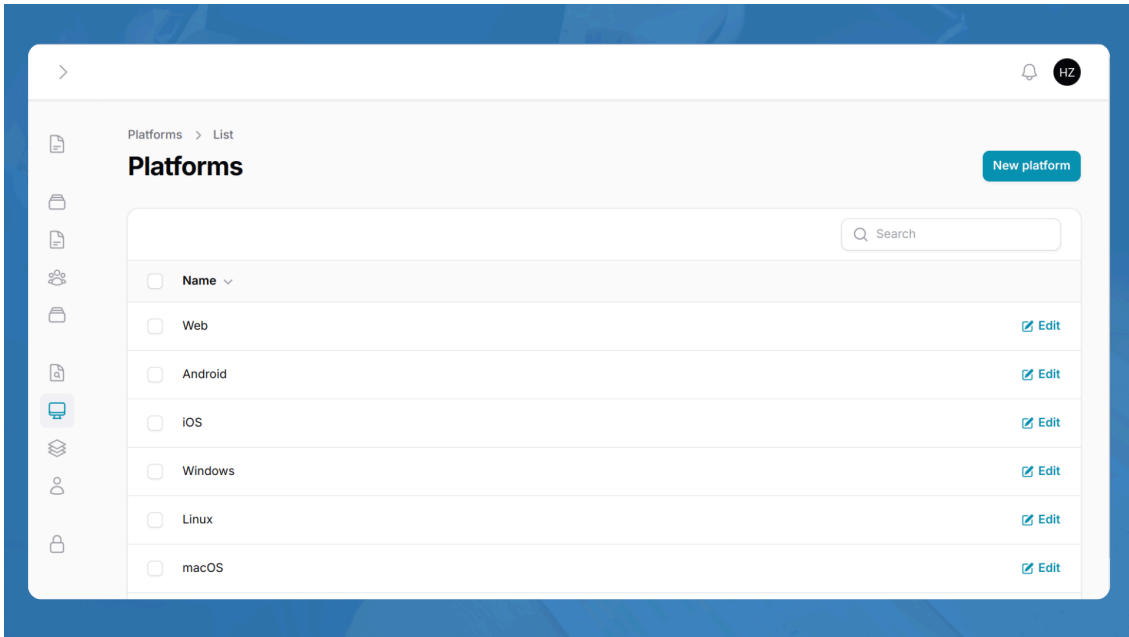


FIGURE 24 - Platform

Figure 24 illustrates the platforms management screen, where users can define and manage different platforms such as web, Android, or iOS.

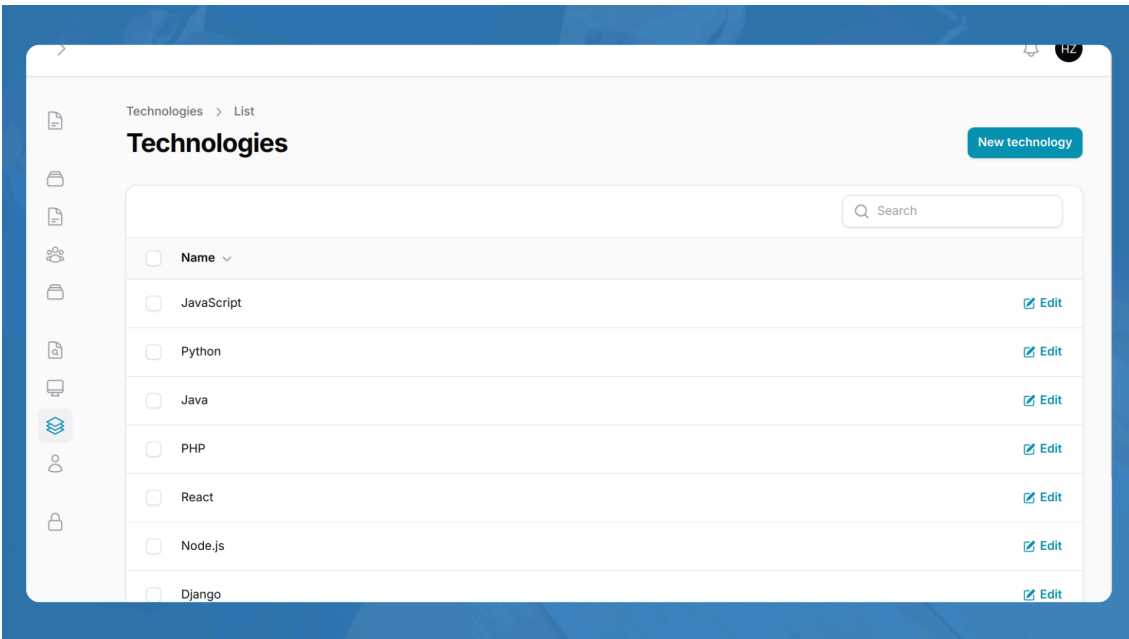


FIGURE 25 - Technologies

Figure 25 shows the technologies management screen, which enables users to define and maintain a list of technologies used within the system, such as programming languages, frameworks, or tools.

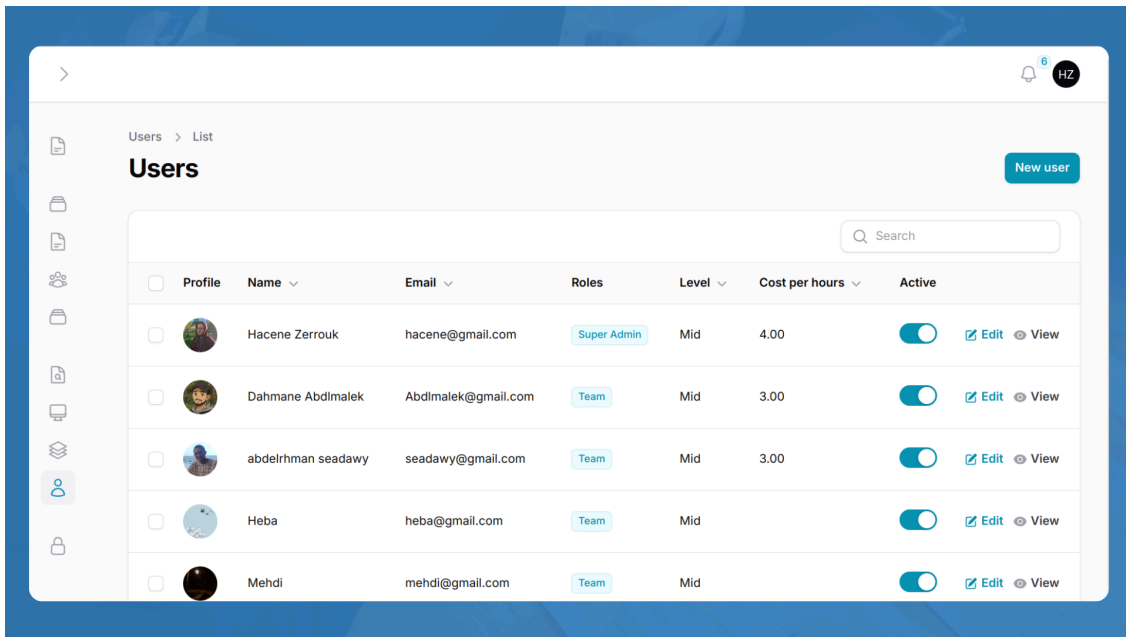


FIGURE 26 - Users

Figure 26 presents the user management screen, which allows administrators to view, add, edit, and remove users from the system. This screen supports role assignment, access control, and user status management, ensuring that the right permissions are granted based on responsibilities within the organization.

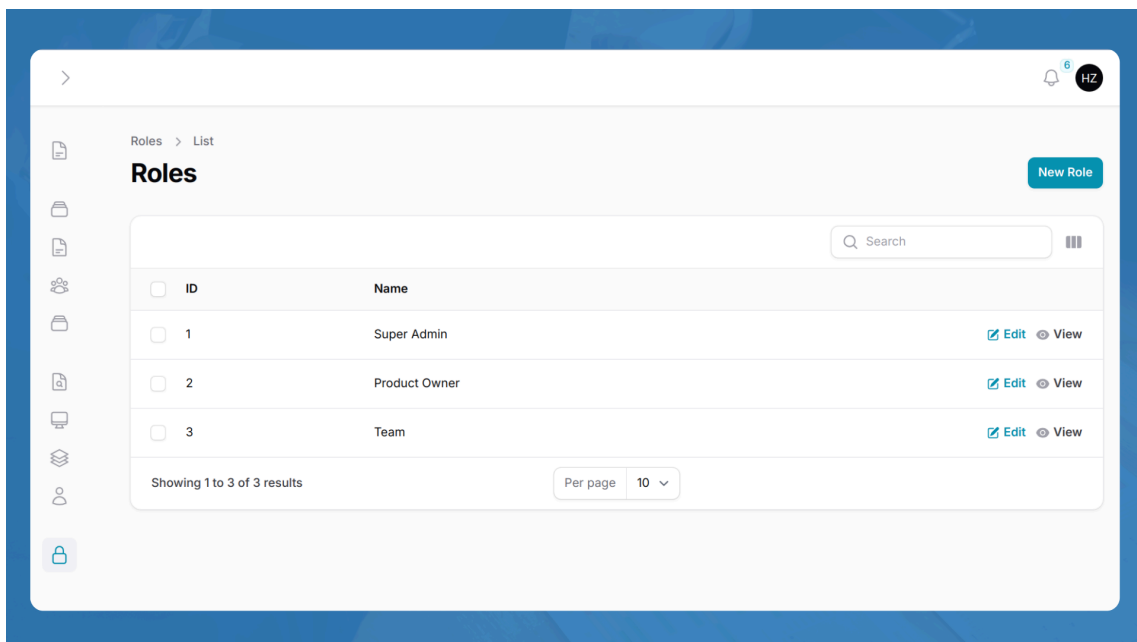


FIGURE 27 - Roles

Figure 26 displays the role management screen, where administrators can create, edit, and assign roles within the system. Each role defines a set of permissions that control access to various features and

actions. This screen ensures that users have appropriate access levels based on their responsibilities, supporting secure and structured system usage.

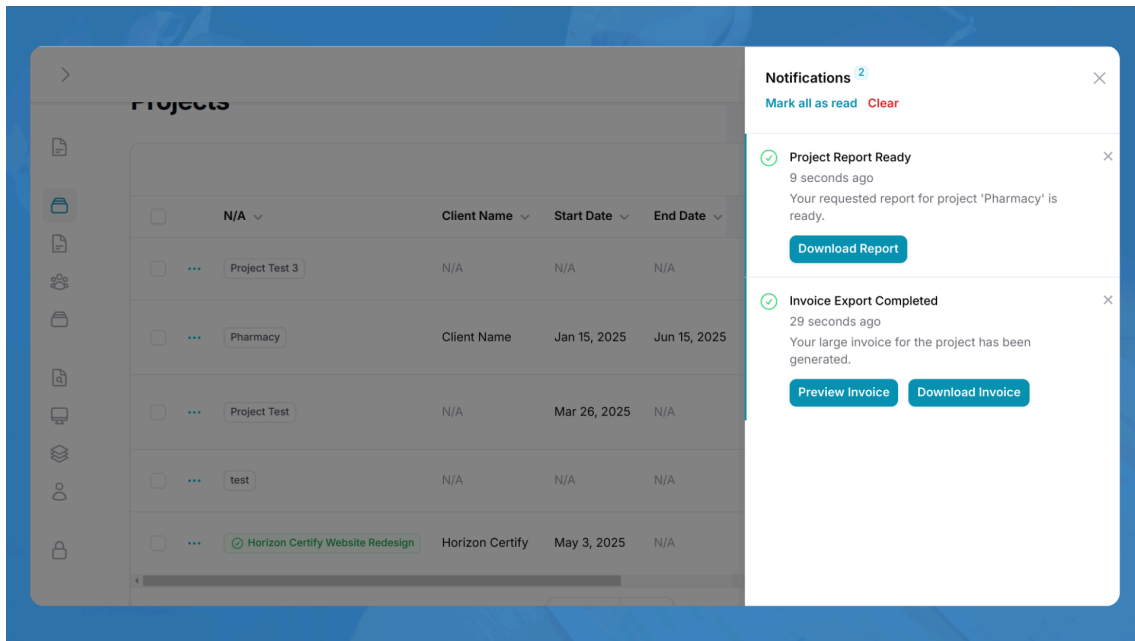


FIGURE 29 - Notifications

Figure 29 presents the notification screen, where users can view incoming notifications generated by the system.

General conclusion

Effective requirements management is a cornerstone of successful software development, yet it remains one of the most persistent and costly challenges in the industry. As demonstrated through both literature and case studies from Five Angles and StackInTech, hidden, incomplete requirements frequently lead to project delays, budget overruns, and compromised outcomes. These challenges are deeply rooted in manual, error-prone processes and inadequate communication during early project stages.

To address this systemic issue, this thesis proposes a transformative solution: an AI-powered Requirement Management Application that automates and enhances key aspects of the requirements engineering lifecycle. By allowing users to start with simple inputs and letting AI generate comprehensive user stories, acceptance criteria, and development tasks, the solution shifts the human role from creation to validation. This approach not only reduces cognitive load but also increases accuracy, clarity, and alignment between teams and stakeholders.

The proposed tool represents a practical response to recurring failures in software projects caused by poor requirements. It aims to improve agility, minimize risks, and raise the overall quality and success rate of software development efforts, ultimately bridging the gap between evolving stakeholder needs and development execution.

References

- [1] History: The Agile Manifesto. consulted on 27/03/2025. URL:
<https://agilemanifesto.org/history.html>
- [2] Agile Project Management. consulted on 27/03/2025. URL:
<https://open.ocolearnok.org/informationssystem/chapter/chapter-13-agile-project-management/>
- [3] The Origin of Scrum. consulted on 27/03/2025. URL:
<https://www.visual-paradigm.com/scrum/what-is-the-evolution-of-scrum>
- [4] Agile Scrum Framework: Values, Roles, Events, Artifacts, and History. consulted on 15/04/2025.
URL:
<https://www.jile.io/blogs/agile-scrum-frawmework-values-roles-events-artifacts-history>
- [5] What is Scrum? An overview of Scrum and The Agile Journey. consulted on 15/04/2025. URL:
<https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview>
- [6] Project estimation: methods and best practices. consulted on 15/04/2025. URL:
<https://www.atlassian.com/work-management/project-management/project-estimation>
- [7] Practical project management book by dmytro nizhebetskyi. consulted on 19/04/2025. URL:
<https://itpmschool.com/practical-project-management-book-offer/>
- [8] What are the most critical Challenges in Requirements Management? consulted on 19/04/2025.
URL:
<https://optimizory.com/blog/rmsis/what-are-the-most-critical-challenges-in-requirements-management>
- [9] Agile Scrum Framework: Values, Roles, Events, Artifacts, and History. consulted on 19/04/2025.
URL:
<https://www.jile.io/blogs/agile-scrum-frawmework-values-roles-events-artifacts-history>
- [10] What is Scrum? An overview of Scrum and The Agile Journey. consulted on 19/04/2025. URL:
<https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/>
- [11] Naming the Pain in Requirements Engineering Contemporary Problems, Causes, and Effects in Practice by D. Mendez Fernandez et al. consulted on 25/04/2025. URL:
<https://arxiv.org/pdf/1611.10288>
- [12] Story points: Estimation guide for user stories in Agile. consulted on 25/04/2025. URL:
<https://asana.com/resources/story-points>
- [13] What Is Story Point Estimation?. consulted on 25/04/2025. URL:
<https://resources.scrumalliance.org/Article/story-point-estimation>
- [14] What is Docker?. Consulted on 04/05/2025. URL:
<https://www.docker.com/resources/what-container>
- [15] OVHcloud - Discover our cloud solutions. Consulted on 04/05/2025. URL:
<https://www.ovhcloud.com>

[16] Filament - Admin panel for Laravel. Consulted on 04/05/2025. URL: <https://filamentphp.com>

[17] React – A JavaScript library for building user interfaces. Consulted on 08/05/2025. URL: <https://reactjs.org>

[18] Git - About Version Control. Consulted on 08/05/2025. URL: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

[19] PHP: What is PHP?. Consulted on 08/05/2025. URL: <https://www.php.net/manual/en/intro-what-is.php>

[20] About GitHub and Git. Consulted on 08/05/2025. URL: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>

[21] MySQL :: MySQL 8.0 Reference Manual. Consulted on 08/05/2025. URL: <https://dev.mysql.com/doc/>

[22] Jira Software – Project Management Tool. Consulted on 08/05/2025. URL: <https://www.atlassian.com/software/jira>

[23] Visual Studio Code - Code Editing. Redefined. Consulted on 08/05/2025. URL: <https://code.visualstudio.com>

[24] OpenAI – Research and deployment of safe artificial general intelligence. Consulted on 08/05/2025. URL: <https://openai.com>

[25] Laravel - The PHP Framework for Web Artisans. Consulted on 08/05/2025. URL: <https://laravel.com/docs>

[26] The Number One Cause for Project Failure and How to Avoid It. Consulted on 27/06/2025. URL: <https://www.astadia.com/blog/project-failure>

الملخص

تركز هذه الأطروحة على تطوير تطبيق إدارة المتطلبات المدعوم بالذكاء الاصطناعي لمعالجة المتطلبات الناقصة والمخفية في تطوير البرمجيات. تظهر الأبحاث أن 71% من فشل المشاريع البرمجية ينبع من سوء إدارة المتطلبات. دراسات الحالة من Five Angles و StackInTech تظهر تأخيرات في المشاريع تصل إلى 4 أشهر بسبب المتطلبات غير المحددة بوضوح.

الحل المقترح يستفيد من OpenAI لتوليد قصص المستخدمين الشاملة ومعايير القبول ومهام التطوير تلقائياً من أوصاف الملاحم البسيطة. النظام المبني على الويب مطور باستخدام Laravel و React و Inertia.js. ويتميز بالتحكم في الوصول القائم على الأدوار وقدرات التصدير السلسلة إلى Jira. يحول التطبيق الدور البشري من إنشاء المتطلبات يدوياً إلى التحقق من المخرجات المولدة بالذكاء الاصطناعي، مما يقلل الحمولة المعرفية ويحسن الدقة.

يُظهر هذا البحث التكامل الفعال للذكاء الاصطناعي في عمليات التطوير الرشيق، مساهماً في هندسة المتطلبات من خلال تقليل مخاطر المشاريع وتحسين معدلات النجاح عبر الأتمتة الذكية.

الكلمات المفتاحية: الذكاء الاصطناعي، هندسة المتطلبات، إدارة المشاريع، التطوير الرشيق، Scrum، React، Laravel، OpenAI، قصص المستخدمين، هندسة البرمجيات.

Résumé

Cette thèse de master se concentre sur le développement d'une application de gestion des exigences alimentée par l'IA pour traiter les exigences incomplètes et cachées dans le développement logiciel. Les recherches montrent que 71% des échecs de projets logiciels proviennent d'une mauvaise gestion des exigences. Les études de cas de Five Angles et StackInTech démontrent des retards de projet allant jusqu'à 4 mois dus à des exigences sous-spécifiées.

La solution proposée exploite OpenAI pour générer automatiquement des user stories complètes, des critères d'acceptation et des tâches de développement à partir de descriptions épiques simples. Le système web est construit avec Laravel, React et Inertia.js, offrant un contrôle d'accès basé sur les rôles et des capacités d'export transparentes vers Jira. L'application fait passer le rôle humain de la création manuelle d'exigences à la validation des sorties générées par l'IA, réduisant la charge cognitive tout en améliorant la précision.

Cette recherche démontre une intégration efficace de l'IA dans les processus de développement agile, contribuant à l'ingénierie des exigences en minimisant les risques de projet et en améliorant les taux de réussite grâce à l'automatisation intelligente.

Mots-clés: Intelligence Artificielle, Ingénierie des Exigences, Gestion de Projet, Développement Agile, OpenAI, Laravel, React, Scrum, User Stories, Génie Logiciel.

Abstract

This master's thesis focuses on developing an AI-powered Requirement Management Application to address incomplete and hidden requirements in software development. Research shows that 71% of software project failures stem from poor requirements management. Case studies from Five Angles and StackInTech demonstrate project delays of up to 4 months due to underspecified requirements.

The proposed solution leverages OpenAI to automatically generate comprehensive user stories, acceptance criteria, and development tasks from simple epic descriptions. The web-based system is built with Laravel, React, and Inertia.js, featuring role-based access control and seamless Jira export capabilities. The application shifts the human role from manual requirement creation to validation of AI-generated outputs, reducing cognitive load while improving accuracy.

This research demonstrates effective AI integration in agile development processes, contributing to requirements engineering by minimizing project risks and improving success rates through intelligent automation.

Keywords: Artificial Intelligence, Requirements Engineering, Project Management, Agile Development, OpenAI, Laravel, React, Scrum, User Stories, Software Engineering.