

PROJENTI
PROJECT MANAGEMENT SYSTEM

Table of Abbreviations

Abbreviation	Definition

Table of Glossary

Term	Definition

1. Introduction

1.1. Vision and Objectives

Projecti's vision is to become the leading Software-as-a-Service (SaaS) project management solution for small and medium-sized enterprises (SMEs). Our primary objective is to deliver a centralized platform that simplifies the entire project lifecycle, from initiation and planning to final closure. To achieve this, Projecti will centralize project workflows and enhance team collaboration through a suite of integrated tools. The platform will provide real-time insights via an intuitive dashboard, empowering managers with the data needed to improve resource allocation, make informed decisions, and ensure projects are successfully delivered on time and within budget.

The initial release will focus on an objective: to validate the core functionalities that are most essential for project and task management. This launch will target a select group of early adopters to prove the concept's value and gather critical feedback. This feedback will directly inform and guide future product iterations and enhancements.

1.2. Problem statement

Small and medium-sized enterprises (SMEs) frequently struggle with project management due to a reliance on separated tools, such as spreadsheets, email, or basic to-do list applications. This fragmentation leads to several critical issues:

- **Lack of Real-Time Visibility:** Without a centralized system, managers cannot get a clear, up-to-the-minute overview of project progress, making it difficult to identify bottlenecks or risks.
- **Inefficient Communication and Collaboration:** Information is scattered across different platforms, causing miscommunication, delays in feedback, and difficulty in tracking decisions.
- **Poor Resource Allocation:** It becomes challenging to monitor team workload and allocate resources effectively, often resulting in burnout for some team members and underutilization for others.
- **Increased Risk of Delays and Budget Overruns:** These inefficiencies collectively contribute to missed deadlines and increased costs, directly impacting the profitability and success rate of projects.

While enterprise-grade project management solutions exist, they are often too complex, feature-heavy, and expensive for SMEs, who typically operate with limited budgets and lack dedicated IT support. There is a clear market need for a solution that is powerful yet simple, affordable, and specifically tailored to the workflow of SMEs.

1.3. Target audience

Projecti's primary target audience is Small and Medium-sized Enterprises (SMEs), typically with 10 to 200 employees, who need a structured project management tool without the high cost and complexity of enterprise software. The platform is designed to serve the developers within these organizations.

Specifically, it empowers the control of Project Manager by providing essential tools for planning, task assignment, progress monitoring, and reporting, giving them clear oversight of the entire project lifecycle. On the other hand, it caters to other team member by offering a simple, intuitive interface to view their assigned tasks, update their status, and collaborate effectively with colleagues.

1.4. MVP Scope-Defining Methodology

The methodology for defining Projecti's MVP scope is guided by Lean Startup principles, which prioritize delivering maximum value and validated learning with minimal effort. Our primary goal is implementing a version that solves the most critical problems for our target audience, enabling us to test our fundamental business hypotheses in a real-world environment.

To prioritize features, we rely on a clear set of criteria to guide the entire development process. We focus on selecting essential features that create a solid foundation for project management and have the greatest impact on solving the most important user problems. This selection is balanced with development feasibility, ensuring we can deliver these features quickly, within budget and minimizing risk from the start.

1.5. Risk and Mitigation

The project relies on several third-party integrations which require access permission to repositories, CI/CD pipelines and monitoring tools. Moreover, a key risk is that changes to project requirements or scope may cause delays. To mitigate this, any modifications to the project scope would require an amended project plan and timeline. Additionally, issues outside of our control would incur additional costs. Clear communication and a formal change request process will be crucial to manage these risks and ensure the project stays on track.

1.6. Limitation

The MVP is designed to be a usable version for development teams to plan projects, track tasks, and begin integrating DevOps tools. This initial version will also include a “lightweight” monitoring dashboard. More advanced features, such as an AI-powered test bot for generating test cases and frameworks, are explicitly planned as future developments and are not included in this MVP.

1.7. MVP Scope

Considering the scope defining methodology, aligning with lean start up strategy and business goals, This MVP will offer the following functions:

1.7.1. Project management

The core of the MVP is a robust project management module designed to provide a centralized hub for all project-related activities. This function will enable leaders to create projects, define key objectives, and break down work into manageable tasks. Team members can then be assigned these tasks, with clear due dates and priorities. The primary interface will be a visual Kanban board, allowing for intuitive status updates. This foundational feature is designed to bring immediate structure and clarity to team workflows, ensuring everyone is aligned on responsibilities and progress.

1.7.2. Test scenario management

To tightly integrate quality assurance into the development lifecycle, the MVP will include a lightweight test scenario management module. This feature will provide a simple interface for QA personnel to organize, and track the status of test cases. A key point will be the ability to link specific test scenarios directly to project tasks. This scenario management creates clear traceability between requirements, development work, and testing efforts, ensuring that quality is considered throughout the entire process.

1.7.3. Metric dashboard

The MVP will feature a simple yet effective metric dashboard to provide developers with an at-a-glance overview of server health. This dashboard will not be overly complex, focusing instead on key server-level indicators. Initial metrics will include visualizations for CPU and memory utilization, API response times, and error rates. The primary goal of this feature is to enable proactive infrastructure management, allowing the team to detect performance issues, anticipate potential failures, and ensure a reliable and seamless experience for all early adopters.

1.7.4. Feedback collection

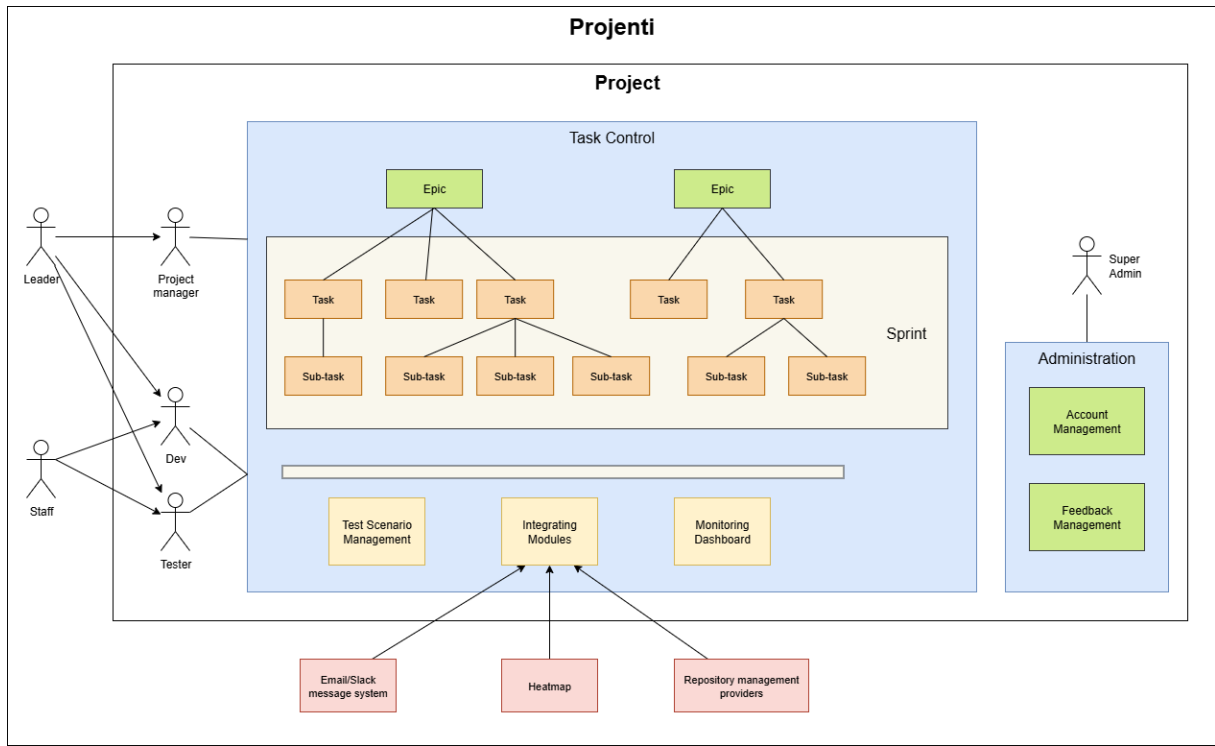
A crucial component of our lean methodology, the MVP will incorporate an integrated feedback collection tool. This feature will offer a direct and seamless channel to communicate between administrators and developers. A template of questions would be created by administrators and sent to project members. All feedback will be captured and organized in a centralized location, providing administrators with invaluable qualitative data to validate working environments and collaborations between project developers.

1.7.5. Third-party integration

The MVP version of this project will include a foundational integration with a critical third-party service. The initial focus will be on a key tool commonly used in development workflows, such as a source control repository like GitHub or GitLab. This integration will allow for basic connectivity, such as linking code commits to specific project tasks. The goal is to demonstrate the platform's ability to connect with external tools, thereby reducing context-switching and streamlining the team's overall workflow.

2. System Architecture

2.1. System Overview



2.2. User roles and responsibilities

The system architecture of Projecti defines three primary user roles at system level: Super Admin, Leader, and Staff.

Actor name	Description	Features
Super Admin	Highest-level administrative actor – this role takes responsible for system-wide setting, managing the accounts of all users.	Account Management
		Feedback Management
Leader / Project Manager	The only actor with authority to create new projects. Upon creating a project, this actor automatically become its designated Project Manager.	Project Initiation
		Project Member Management
		Project Overview
Project Members	This is a general role for any user assigned to a project by a Project Manager. Project Members are responsible for executing the tasks assigned to them and collaborating with the team.	Project Component Creation
		Component Status Modification

Actor name	Description	Features
		Monitoring Dashboard
		Third-party Repository Account Connection
Tester	A specialized role assigned to a Project Member by the Project Manager. This actor is responsible for quality assurance tasks, such as running test scenarios.	Test Scenario Management

2.3. Key features

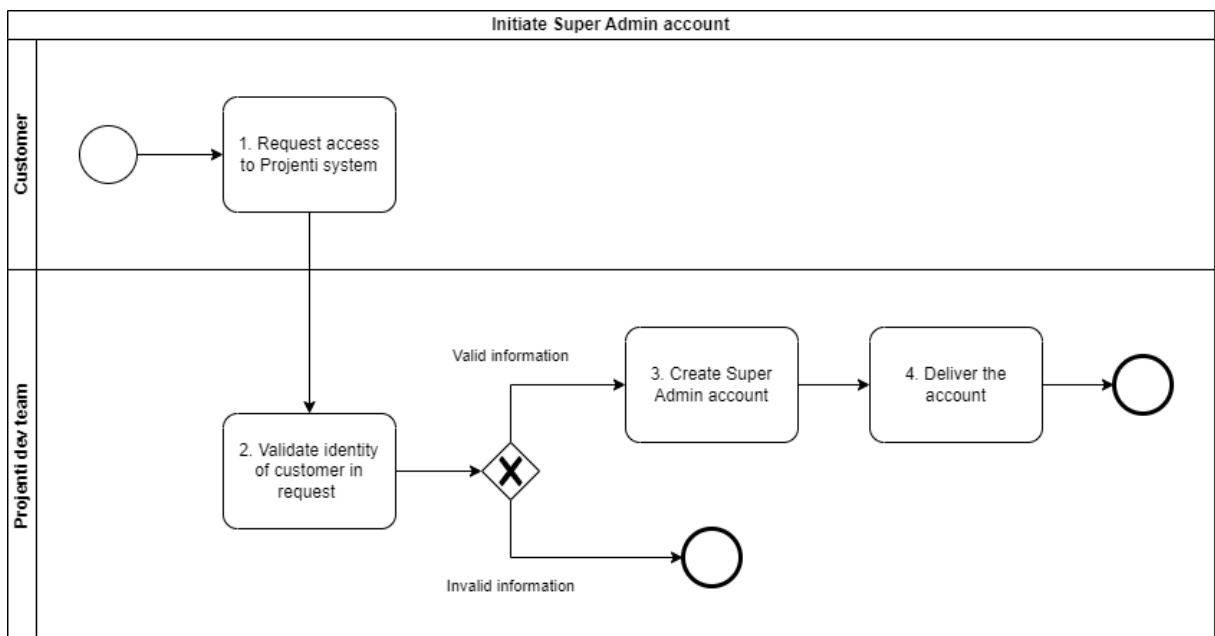
2.3.1. General management

2.3.1.1. Use case 01 – Super Admin – Initiate admin account

a. General information

The platform is operated by the assumption that Projenti succeeds in acquiring their first customer. As the result, the customer will nominate one of their employees as the system admin to create the accounts on behalf of their organization. This Super Admin account will be the main account and perform all system-level actions as briefly described at “2.2. User roles and responsibilities”.

b. Workflow



c. Feature description

Use case Name	Initiate admin account
Primary Actor	Super Admin

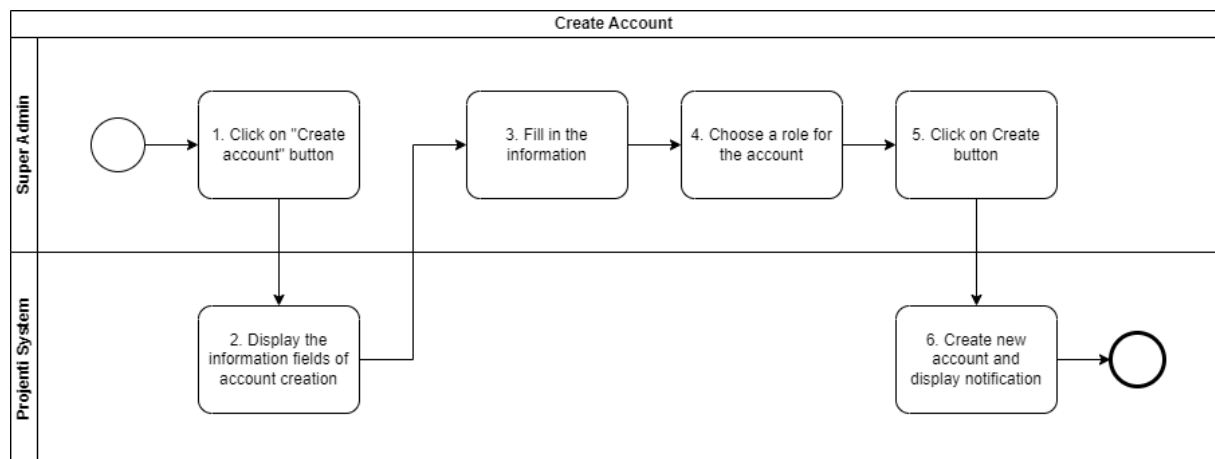
Trigger	Super Admin activates the system
Pre-Condition(s)	Both organizations have finished the required documents and signing the contract
Basic Flow	1. Customer request access to Projenti system 2. Projenti dev team validates the identity of customer in request message (such as digital signature, ...) 3. Projenti dev team creates Super Admin account 4. Projenti dev team delivers the account through a security channel <i>End</i>
Alternative Flow(s)	<i>The identity validation of customer is failed</i> 3a1. Projenti dev team rejects the account creating request <i>End</i>
Post-Condition(s)	Super Admin account is created and could be used to log in

2.3.1.2. Use case 02 – Super Admin – Create account

a. General information

The Super Admin account could add new Leader/Staff account into the system to perform their company tasks. The quantity of non-admin accounts must be aligned with the number of user license signed by Projenti and their customers.

b. Workflow



c. Feature description

Use case Name	Create account
Primary Actor	Super Admin
Trigger	Super Admin clicks on the “Create account”
Pre-Condition(s)	Super Admin has logged in Projenti system

Basic Flow	<ol style="list-style-type: none"> 1. Super Admin clicks on “Create account” button 2. System displays the information fields of account creation 3. Super Admin fills in the information of the account 4. Super Admin chooses a role for the account 5. Super Admin confirms and clicks on “Create” button 6. System creates new account and displays notification <p><i>End</i></p>
Alternative Flow(s)	<p><i>The filled information of the account is invalid</i></p> <ol style="list-style-type: none"> 6a1. System alerts the fields with invalid information 6a2. Super Admin reviews and changes the input data <p><i>Continue at step 5</i></p>
Post-Condition(s)	<p>The newly created account could be used to login</p> <p>A new account should be displayed in account list of Super Admin</p>

2.3.1.3. Use case 03 – Super Admin – Update account status

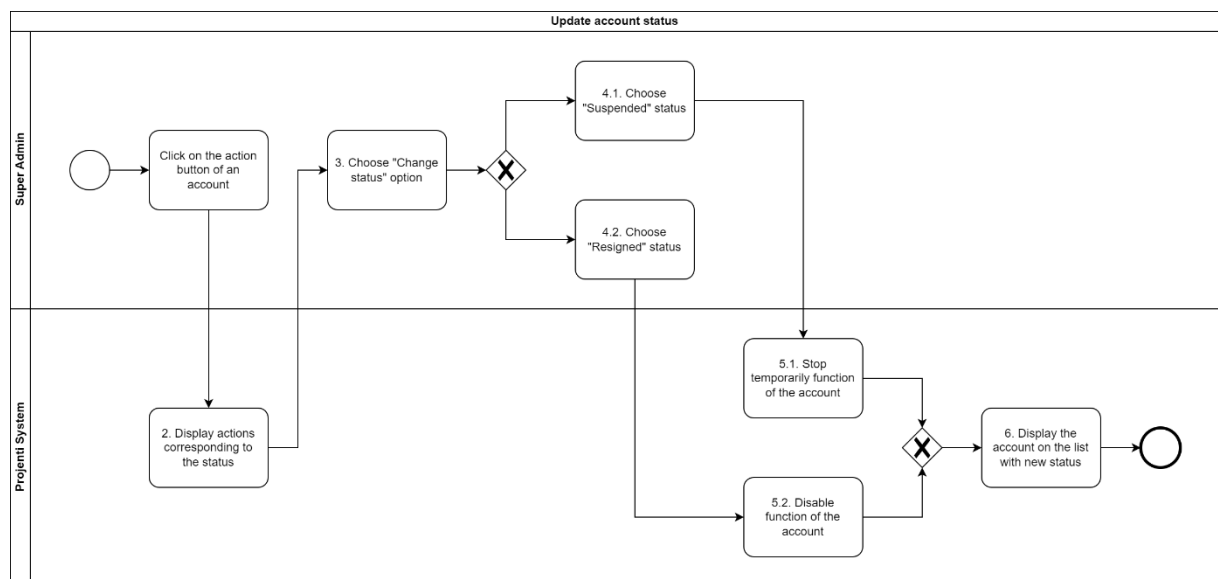
a. General information

Super Admin could update the chart of staff account status in line with their work status.

If an account is marked with status “Suspended”, this account could be reactivated and the slot for account creation in license of company is still occupied.

If an account is marked with status “Resigned”, this account couldn’t be reactivated and the slot for account creation in license of company is freed, which allows Super Admin to create a new one.

b. Workflow



c. Feature description

Use case Name	Update account status
----------------------	------------------------------

Primary Actor	Super Admin
Trigger	Super Admin clicks on Action column of an account
Pre-Condition(s)	Super Admin has signed in The chosen account must be in “Active” status
Basic Flow	1. Super Admin clicks on action button of an account 2. System displays actions corresponding to the status of the account 3. Super Admin chooses “Change status” option 4.1. Super Admin chooses status “Suspended” 5.1. System suspends the working functions of the account until Super Admin reactivates them 6. System displays the account on the list with changed status <i>End</i>
Alternative Flow(s)	4.2. Super Admin chooses status “Resigned” 5.2. System disables the working functions of the account <i>Continue at step 6</i>
Post-Condition(s)	Account with “Suspended” and “Resigned” status could not be used to log in to Projecti system

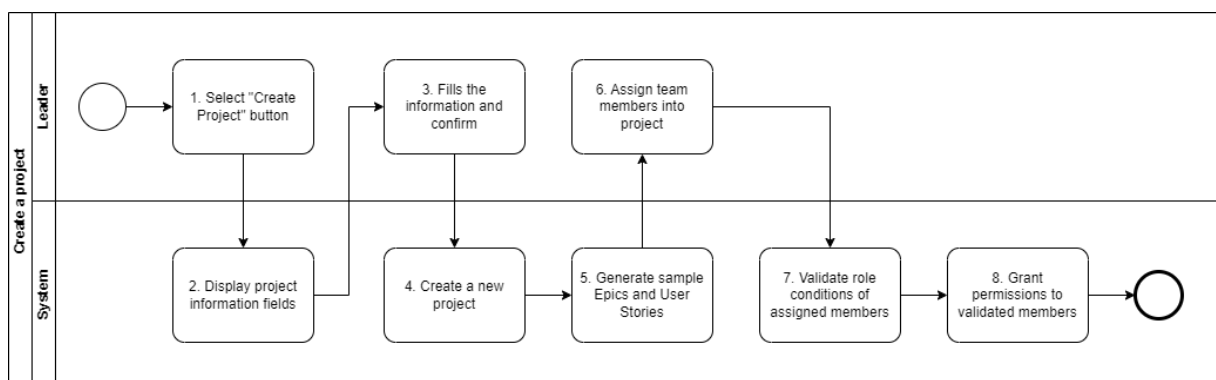
2.3.2. Project management

2.3.2.1. Use case 04 – Leader – Create project

a. General information

This activity establishes the initial workspace for planning, task management, and team collaboration. A new project is only created by Leader, then a sample epic and user stories would be generated automatically. Subsequently, the Leader, who created the project, could assign other members with corresponding roles as Project Manager, Developer or Tester.

b. Workflow



c. Feature description

Use case Name	Create project
----------------------	----------------

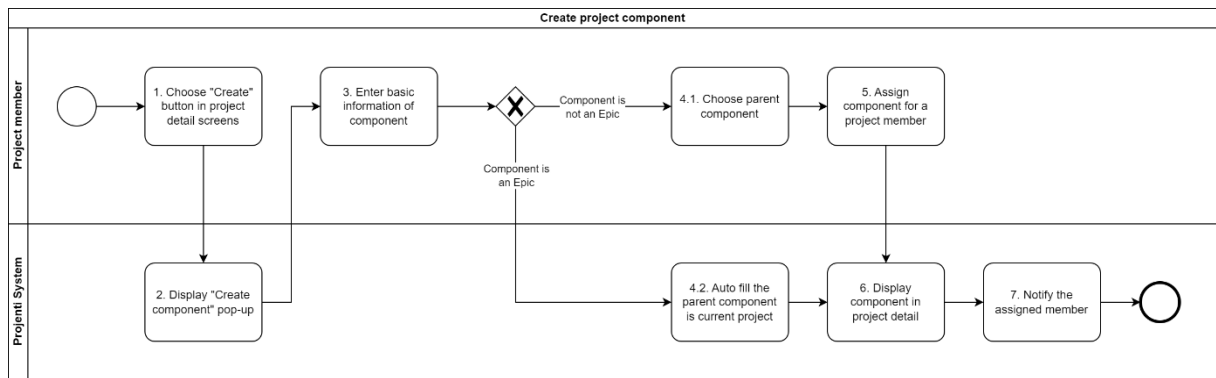
Primary Actor	Leader
Trigger	Leader clicks on the “Create Project” button
Pre-Condition(s)	Leader has signed in
Basic Flow	<ol style="list-style-type: none"> 1. Leader clicks on the “Create Project” button 2. System opens the “Project Information” screen 3. Leader fills in the information of the project 4. System validates, creates samples of an epic and user stories 5. System opens the “Team member” screen 6. Leader selects team members from the list and assign project role for each of them 7. System validates the role condition of assigned members 8. System saves the list of team members 9. System displays the screen <p><i>End</i></p>
Alternative Flow(s)	<p><i>System detects the invalid in role conditions of assigned members</i></p> <ol style="list-style-type: none"> 8a1. System alerts the invalid assigned members 8a2. Leader re-selects members for the project <p><i>Continue at step 7</i></p>
Post-Condition(s)	<p>The project is created with provided information</p> <p>The assigned member of the project is listed in the detail section</p>

2.3.2.2. Use case 05 – Project member – Create a project component

a. General information

In order to provide a powerful and hierarchical work breakdown structure, the highest container in Projecti system is a Project. A Project include numerous Epics which stands for large-scale features or major phases of work. Each Epic could be broken down into specific actions known as Task and assigned them to members of the project. Sometimes, a Task remains high complexity and requires multiple members to resolve, therefore, it could be further divided into smaller steps called Sub-task. Every member has full authority to create a component and assign it (task, sub-task) to another member of the project.

b. Workflow



c. Feature description

Use case Name	Create a project component
Primary Actor	Project members
Trigger	Project member chooses “Create” button in project detail screen
Pre-Condition(s)	Staff account must be included into a project by its creator
Basic Flow	1. Member clicks on the “Create” button in project detail screen 2. System displays “Create component” pop-up 3. Member enters basic information of the component (type, name, description, start date – end date, ...) <i>In case the component is not an Epic</i> 4.1. Member chooses parent component for the current one 5. Member assigns this component to a project member including themselves 6. System displays component in project detail screen 7. System notifies the assigned member (email, notification, ...) <i>End</i>
Alternative Flow(s)	<i>In case the component is an Epic</i> 4.2. System auto-fill the parent component of the Epic is current project <i>End</i>
Post-Condition(s)	The component is displayed in project detail screens The assigned member receives the notification of newly assigned task

2.3.2.3. Use case 06 – Update task status

a. General information

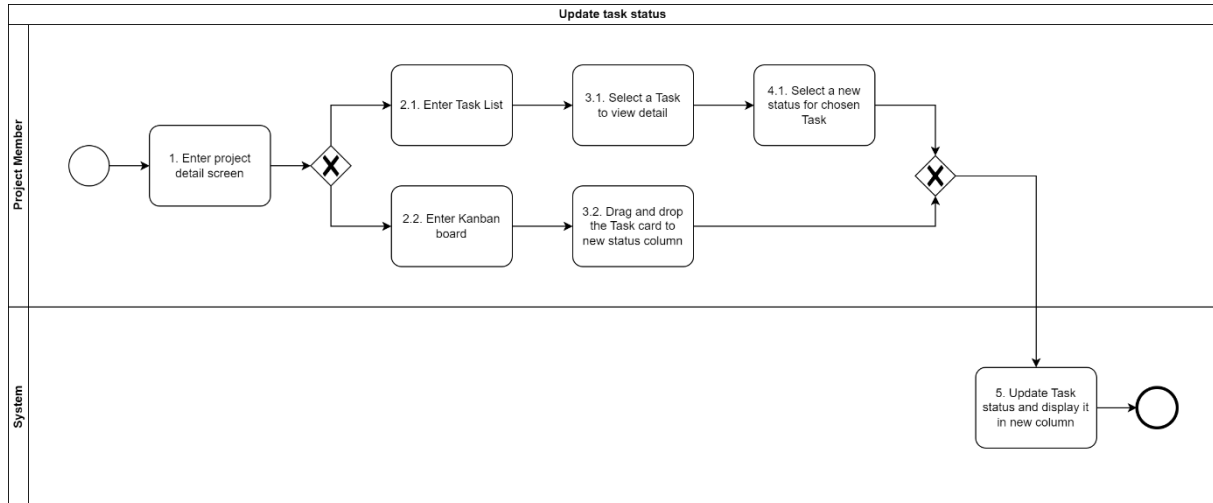
Updating task status is a fundamental action in Projenti that provides real-time visibility into project momentum. There are two methods for updating the progress of a task.

The primary method is using Kanban board, where users can “drag and drop” or reposition a task card between workflow columns, such as from "To-Do" to "In Progress," and finally to

"Done". The secondary method is changing the status in the task detail screen. This action is accomplished by selecting a task to open a comprehensive view, where the user can manually change the status using a designated dropdown menu.

Both mechanisms provide transparent information to ensure that project status is accurately synchronized for all team member.

b. Workflow



c. Feature description

Use case Name	Update task status
Primary Actor	Project members
Trigger	Project member moves a task card into new column or Project member select a new status for task in detail screen
Pre-Condition(s)	Staff account must be included into a project by its creator Project member has entered project detail screen
Basic Flow	1. Member enters project detail screen <i>In case member edits task status in detail screen</i> 2.1. Member enters task list screen 3.1. Member selects a task to view or edit detail 4.1. Member selects a new status for chosen task 5. System updates the status of chosen task and displays it in the column corresponding to the status <i>End</i>
Alternative Flow(s)	<i>In case member edits Task status in Kanban board</i> 2.2. Member enters Kanban board 3.2. Member drags or repositions a task card into new column <i>Continue at step 5</i>

Post-Condition(s)	<p>The status of a task must be matched with the column</p> <p>The overview statistic must be updated after status change</p>
--------------------------	---

2.3.3. Monitoring dashboard

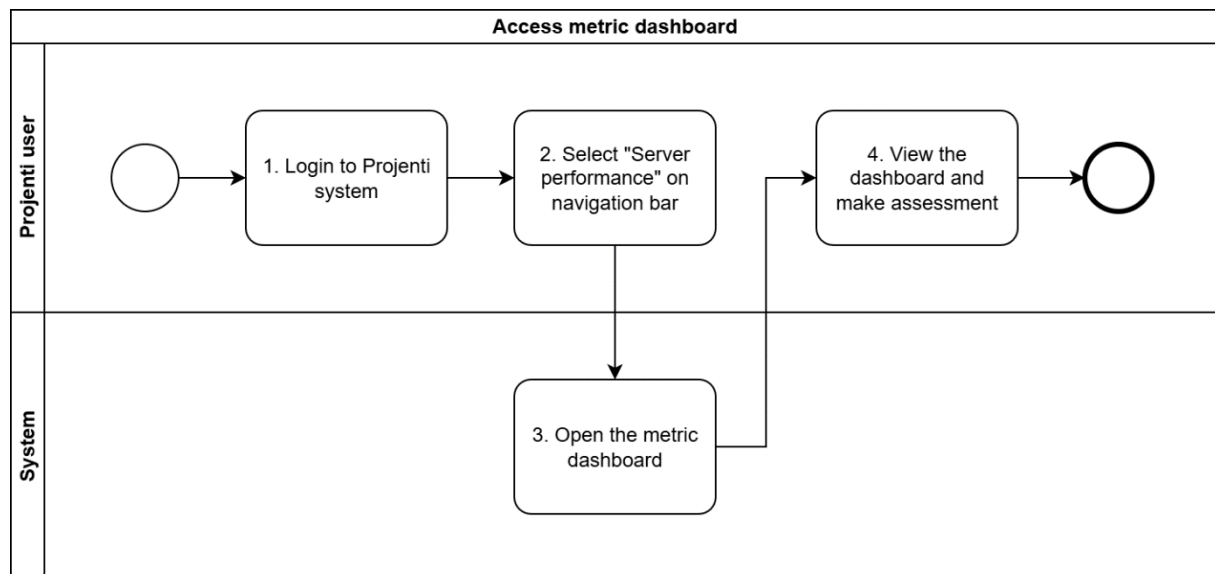
2.3.3.1. Use case 06 – Access metric dashboard

a. General Information

Metric dashboard provides real-time monitoring of key server performance indicators such as CPU usage, memory consumption, disk I/O, network throughput, and uptime. By tracking these metrics on a centralized dashboard, administrators can set alert thresholds, analyze historical data to detect issues early and ensure system reliability.

The dashboard is temporarily available for all Projenti user in MVP version. In case of advanced requirement of information security, the dashboard would be limited to specific account.

b. Workflow



c. Feature description

Use case Name	Access metric dashboard
Primary Actor	Projenti user
Trigger	Projenti user selects “Server performance” on navigation bar
Pre-Condition(s)	Projenti user has signed in
Basic Flow	<ol style="list-style-type: none"> 1. Projenti user logs in Projenti system 2. Projenti user selects “Server performance” on navigation bar 3. System opens the metric dashboard displaying information of server and webservice 4. Projenti user views the dashboard and make assessment <p>End</p>

Post-Condition(s)	The dashboard fetch and update the dashboard at a pre-configured interval
--------------------------	---

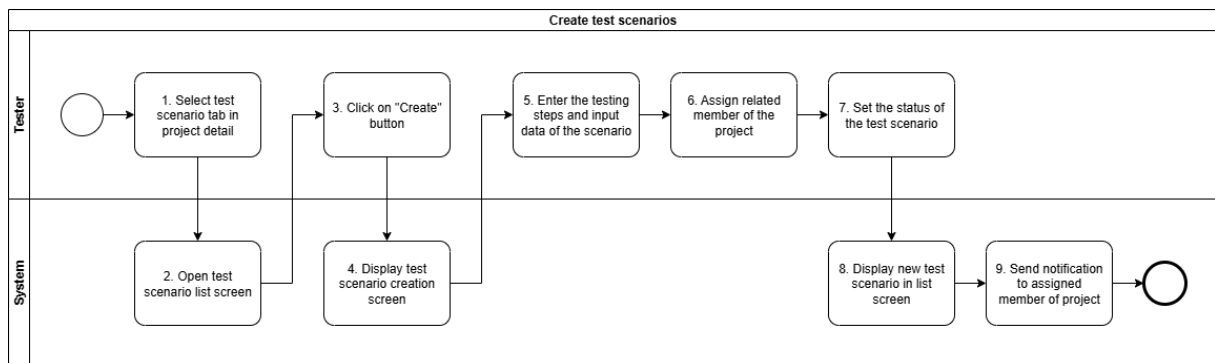
2.3.4. Test scenario management

2.3.4.1. Use case 07 – Create test scenarios

a. General Information

Test Scenarios define end-to-end conditions under which software features are validated, ensuring that the system behaves as expected in real-world use cases and aligned with business requirements. This function allows Tester in every project to outline testing flows, input data, and expected outcomes. These scenarios help project team detect functional gaps, verify business requirements, and improve overall product quality.

b. Workflow



c. Feature description

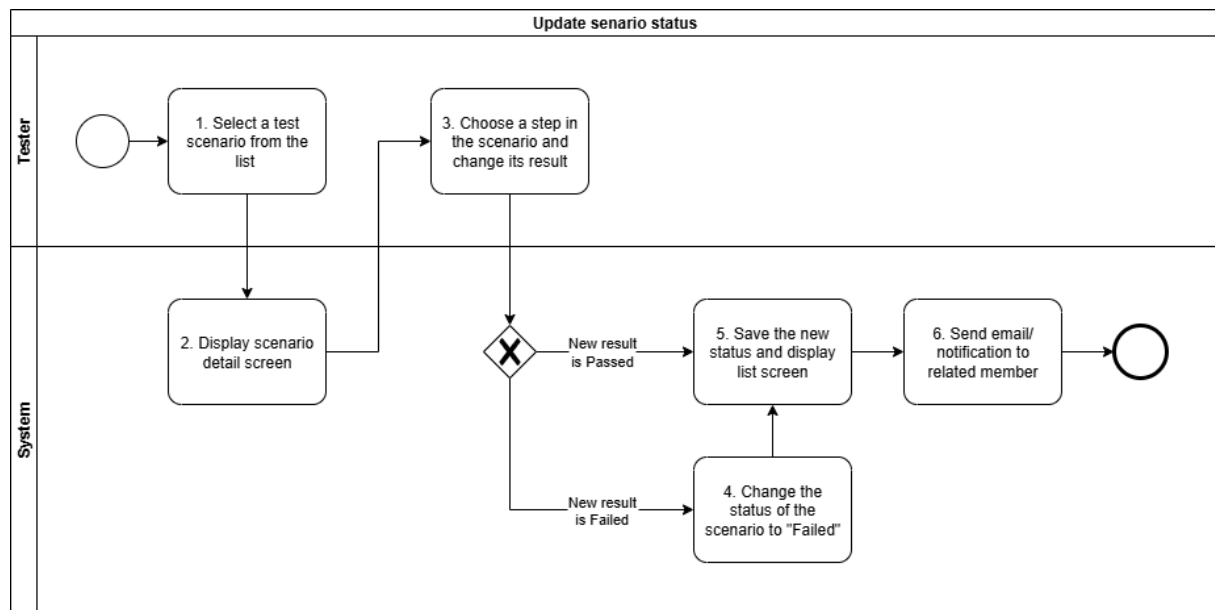
Use case Name	Create test scenarios
Primary Actor	Tester
Trigger	Tester clicks on “Create” button in test scenario list screen
Pre-Condition(s)	Tester has opened detail screen of a project
Basic Flow	<ol style="list-style-type: none"> 1. Tester selects test scenario tab in project detail 2. System opens test scenario list screen 3. Tester clicks on “Create” button 4. System displays test scenario creation screen 5. Tester enters the testing steps and input data of the scenario 6. Tester assigns related member of the project 7. Tester sets the status of the test scenario 8. System displays new test scenario in list screen 9. Send notification to related member <p><i>End</i></p>
Post-Condition(s)	The scenario is displayed in the test scenario list screen

2.3.4.2. Use case 08 – Update scenario status

a. General Information

In a project of Projenti system, testers and project creators could record and modify the execution state of defined test scenarios with multiple status. This functionality ensures that testing progress is accurately tracked, provides visibility by notifying related members via email or message apps. By maintaining up-to-date scenario statuses, the project team can prioritize defect resolution and make decision in planning steps along the development lifecycle.

b. Workflow



c. Feature description

Use case Name	Create test scenarios
Primary Actor	Tester
Trigger	Tester clicks on “Create” button in test scenario list screen
Pre-Condition(s)	Tester has opened detail screen of a project
Basic Flow	1. Tester selects a test scenario from the list screen 2. System displays scenario detail screen 3. Tester chooses a step in the scenario and changes its result (Failed or Passed) <i>In case the new result is Failed</i> 4. System changes the general status of the scenario to “Failed” 5. System saves the new status and displays list screen 6. System sends email or notification to related member of the scenario <i>End</i>

Alternative Flow(s)	<i>In case the new result is Passed Skip step 4 and continue at step 5</i>
Post-Condition(s)	The chosen scenario is displayed in the list with new status All related member of the scenario must receive information via email or internal notification

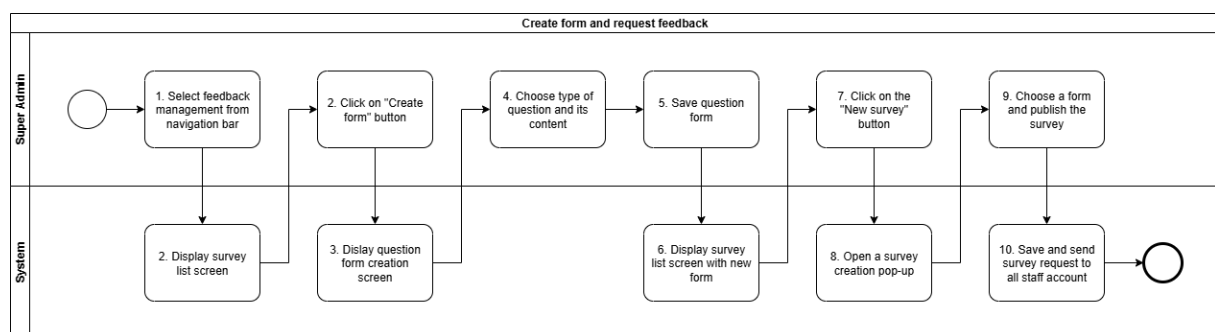
2.3.5. Feedback collection

2.3.5.1. Use case 09 – Create form and request feedback

a. General Information

To enhance the supervision ability, Projenti provides Super Admin with an internal feedback tool, which could send and receive response from other members. In order to create a survey, Super Admin should define questions, configure response option and set a submission deadline. Once published, the form is automatically distributed to selected staff accounts, and the system tracks submission status until the expiration date.

b. Workflow



c. Feature description

Use case Name	Create form and request feedback
Primary Actor	Super Admin
Trigger	Super Admin
Pre-Condition(s)	Super Admin select feedback management function from navigation bar
Basic Flow	<ol style="list-style-type: none"> 1. Super Admin selects Feedback Management from the navigation bar 2. System displays the survey list screen 3. System displays the question form creation screen 4. Super Admin chooses the type of question and its content 5. Super Admin saves the question form 6. System displays the survey list screen with the newly created form 7. Super Admin clicks the "New Survey" button 8. System opens a survey creation pop-up 9. Super Admin chooses a form and publishes the survey

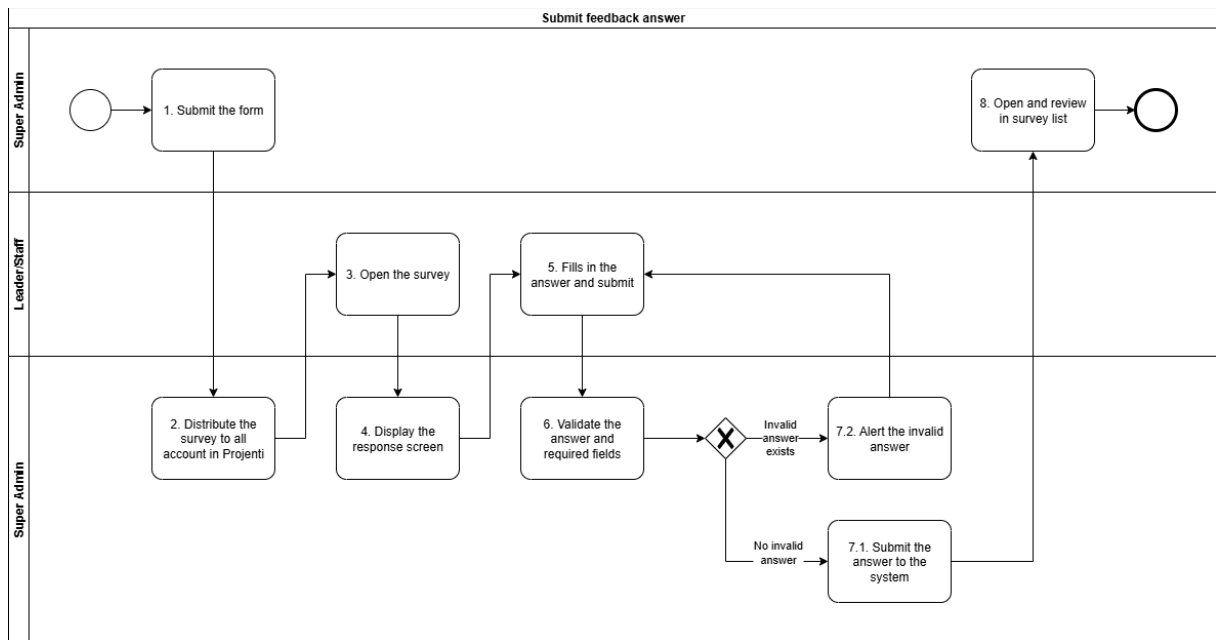
	10. System saves the survey and sends the survey request to all staff accounts
Post-Condition(s)	The chosen scenario is displayed in the list with new status All related member of the scenario must receive information via email or internal notification

2.3.5.2. Use case 10 – Submit feedback answer

a. General Information

After the feedback form has been distributed by Super Admin, each user could provide the answers regarding their tasks or work-related experiences, and submit their responses. Users can save drafts, edit answers before submission, and confirm completion once finalized. The system records each submission and updates the response status in real time for tracking purposes.

b. Workflow



c. Feature description

Use case Name	Submit feedback answer
Primary Actor	All Leader/Staff account of Progenti
Trigger	Super Admin submits the feedback form
Pre-Condition(s)	The survey form has been created
Basic Flow	<ol style="list-style-type: none"> 1. Super Admin submits the feedback form 2. System distributes the survey to all accounts in Progenti 3. Leader/Staff opens the survey 4. System displays the response screen

	5. Leader/Staff fills in the answers and submits them 6. System validates the answers and checks required fields <i>In case no invalid answer exists</i> 7.1. System saves the answers. 8. Super Admin opens and reviews the answers <i>End</i>
Alternative Flow(s)	<i>In case an invalid answer exists</i> 7.2. System alerts the invalid answers for Leader/Staff <i>Continue at step 7.1</i>
Post-Condition(s)	The answers must be recorded and submit in real-time The answers must be aligned with survey form questions

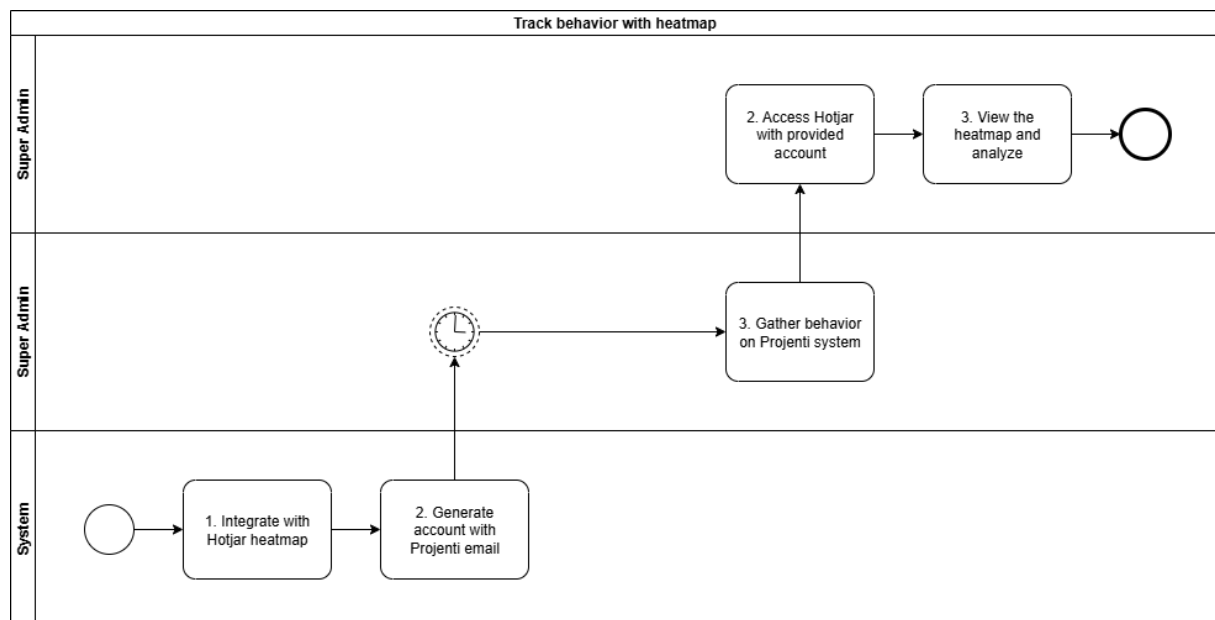
2.3.6. Third-party integration

2.3.6.1. Use case 11 – Track behavior with heatmap

a. General Information

Projecti system empowers administrators to activate Heatmap (provided by Hotjar) on all pages of the application. Once enabled, Hotjar automatically records user interactions such as clicks, scroll depth, and navigation patterns then visualize on their website. Super Admin could use the account which has been generated by Projecti to access the heatmap on Hotjar's website.

b. Workflow



c. Feature description

Use case Name	Track behavior with heatmap
Primary Actor	Super Admin

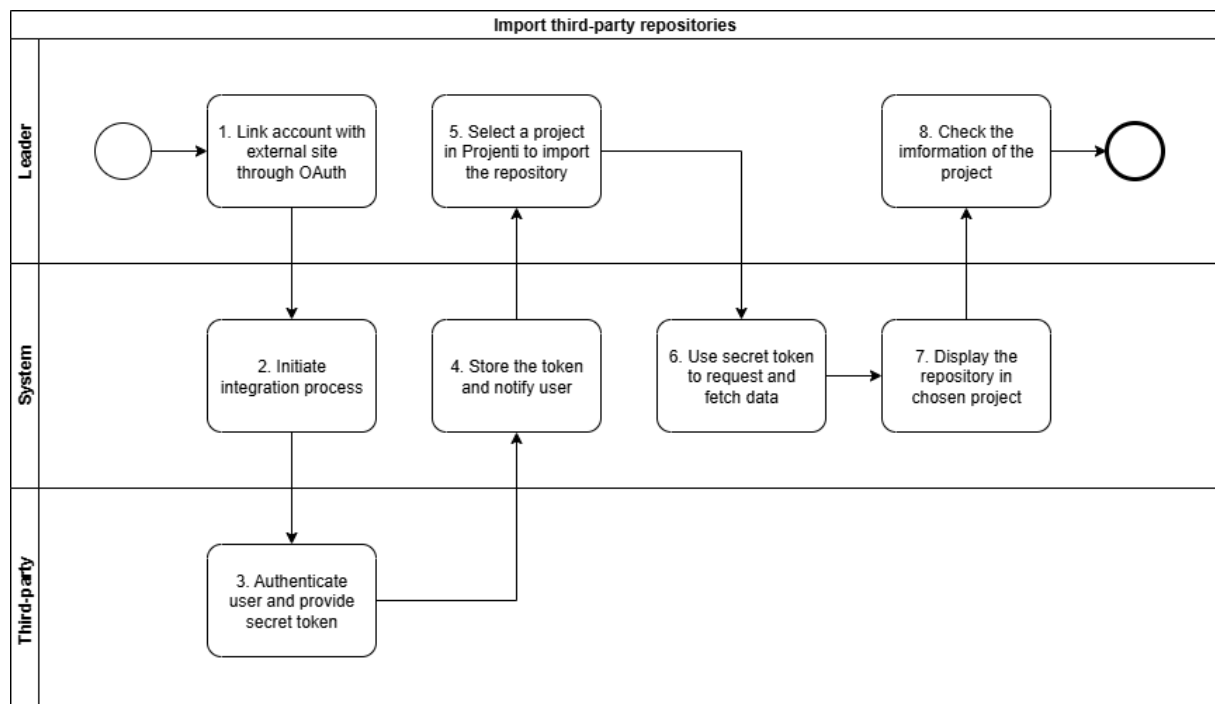
Trigger	Super Admin logs in to Hotjar with provided account
Pre-Condition(s)	Projenti system has integrated with Hotjar heat map An account has been generated for using Hotjar service
Basic Flow	<ol style="list-style-type: none"> 1. System integrates with Hotjar heatmap 2. System generates a Hotjar account using the Projenti email 3. Super Admin gathers user behavior data on the Projenti system (clicks, scrolls, navigation) 4. Super Admin accesses the Hotjar platform with the provided account 5. Super Admin views the generated heatmap and analyzes user behavior insights
Post-Condition(s)	Hotjar could create visualization dashboard from Projenti's URL

2.3.6.2. Use case 12 – Import third-party repositories

a. General Information

In the situation of repository transition, developers could link external source code directly into Projenti. Through OAuth-based authentication, users can securely connect their GitHub, GitLab, or Bitbucket... accounts. Once connected, the system automatically fetches commit history, branch information, and related metadata. The integration also allows synchronization of repository updates and fetch activity information of CI/CD pipeline.

b. Workflow



c. Feature description

Use case Name	Import third-party repositories
Primary Actor	Leader
Trigger	Leader links the account with the external site through OAuth
Pre-Condition(s)	Leader must be the creator of the project Leader must have full-access to the third-party repository
Basic Flow	<ol style="list-style-type: none"> 1. Leader links the account with the external site through OAuth 2. System initiates the integration process 3. Third-party authenticates the user and provides a secret token 4. System stores the token securely and notifies the user of successful authentication 5. Leader selects a project in Projenti to import the repository 6. System uses the secret token to request and fetch repository data 7. System displays the repository in chosen project 8. Leader checks the information of the newly created project
Post-Condition(s)	A new repository has been linked with a project of Projenti The data of the repository (CI/CD pipeline, commit history, ...) could be displayed within new project

2.4. System Mockups

2.4.1. Super Admin

2.4.1.1. Account management screen

LOGO

Accounts

Departments

Feedback

Language

Account Management

Account Management

ENG

Admin Demo

Super Admin

SA

Search

Department

Role

Add Account

Account Name	Account ID	Email	Department	Role	Updated at	Status	Action
Account for demo 01	1544152	Example012345@demo.com.org	Demo council 01	Super Admin	18/09/2024	Working	<div><div></div><div></div></div>
Account for demo 02	14864651	Example012345@demo.com.org	Demo council 01	Project Manager	18/09/2024	Resigned	<div><div></div><div></div></div>
Account for demo 03	16464665	Example012345@demo.com.org	Demo council 01	System Administrator	18/09/2024	Suspend	<div><div></div><div></div></div>
Account for demo 04	11316	Example012345@demo.com.org	Demo council 01	Developer	18/09/2024	Working	<div><div></div><div></div></div>
Account for demo 05	1544855	Example012345@demo.com.org	Demo council 01	Designer	18/09/2024	Suspend	<div><div></div><div></div></div>
Account for demo 06	487878	Example012345@demo.com.org	Quality Control	Super Admin	18/09/2024	Working	<div><div></div><div></div></div>
Account for demo 07	875422	Example012345@demo.com.org	Demo council 01	Super Admin	18/09/2024	Resigned	<div><div></div><div></div></div>
Account for demo 08	1641512	Example012345@demo.com.org	Demo council 01	Super Admin	18/09/2024	Working	<div><div></div><div></div></div>
Account for demo 09	1548979	Example012345@demo.com.org	Demo council 01	Super Admin	18/09/2024	Working	<div><div></div><div></div></div>
Account for demo 10	98781312	Example012345@demo.com.org	Demo council 01	Super Admin	18/09/2024	Resigned	<div><div></div><div></div></div>

Number of rows: 10

Showing 1 to 10 out of 50 records

<

1

2

3

...

5

6

>

2.4.1.2. Account creation screen

LOGO

Accounts

Departments

Feedback

Language

Add New Account

Account Management > Add New Account

ENGAdmin DemoSuper AdminSA

Personal Information

Working Information

First Name*

Last Name*

Date of Birth*

Gender*

Mobile Number

Email*

Address

City

Country

Avatar

Upload

Cancel

Continue

2.4.1.3. Survey and feedback management

Pronjenti

V1.0.0

For you

Recent

Starred

Projects

Project 1

View all projects

Dashboards

Dashboard 1

View all dashboard

Surveys

Surveys

When you resolve a request, users will receive an email asking them to rate and comment on their point of view.

Search survey ...

Create format +

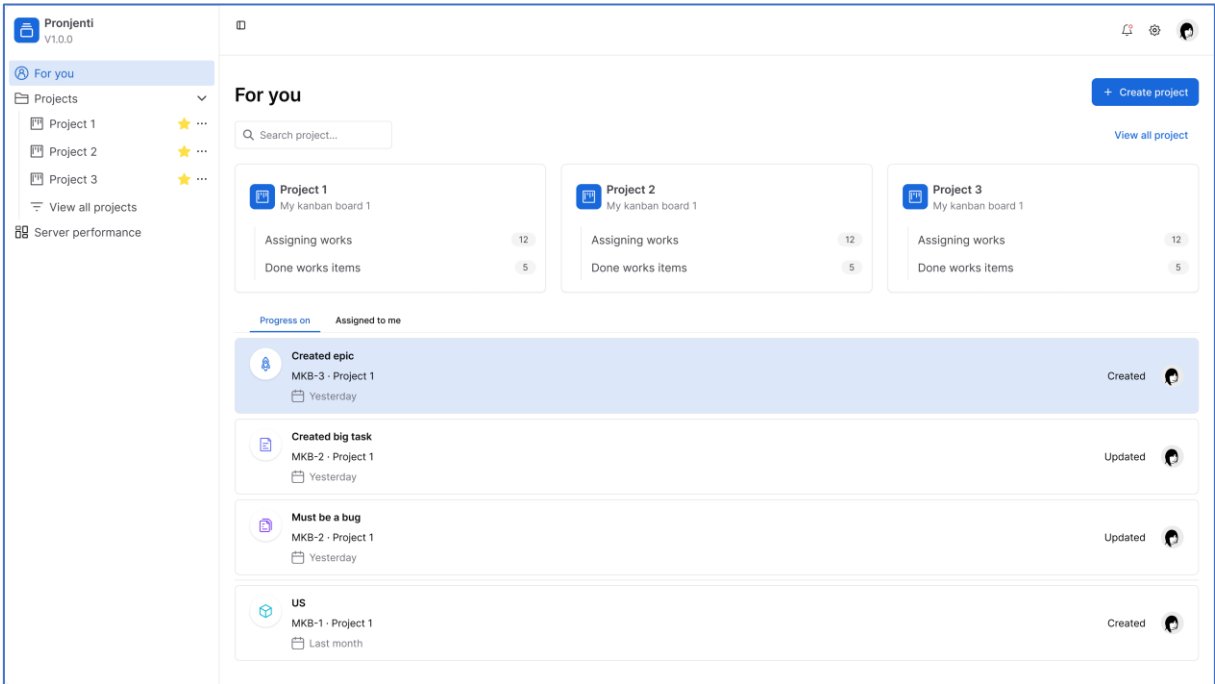
<input type="checkbox"/>	Survey name	Template name	Due date	Response rate	Position	
<input type="checkbox"/>	Check the hygiene status	First template	08/09/2024	80/100	Administration	...
<input type="checkbox"/>	Education and Learning about AI	First template	08/09/2024	65/100	Administration	...
<input type="checkbox"/>	Overtime Policy	First template	08/09/2024	90/100	Administration	...
<input type="checkbox"/>	Survey 123	First template	08/09/2024	85/100	Administration	...

0 of 1 row(s) selected.

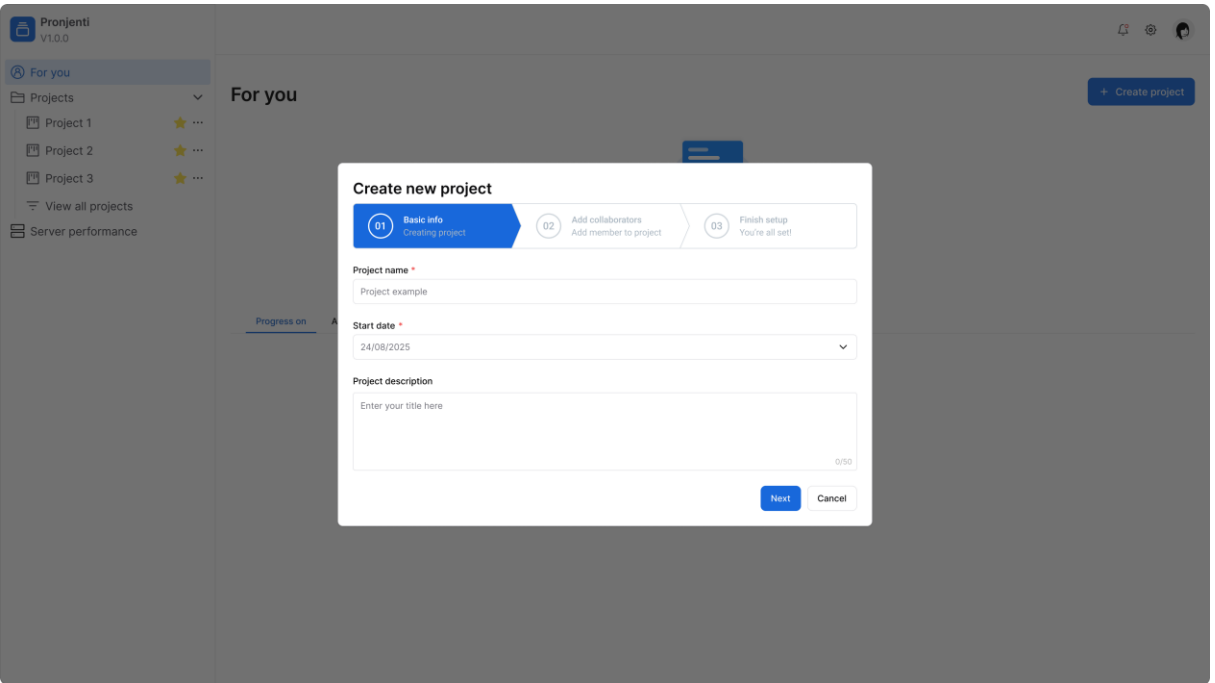
Page 1 of 10

2.4.2. Leader/Staff

2.4.2.1. Homepage screen



2.4.2.2. Project initiation



2.4.2.3. Project member management

Pronjenti

V1.0.0

← Project settings

Project detail

Users

Survey

Projects > Users

Users

Search roles...

Role

Status

+ Invite user

<input type="checkbox"/>	User	Status	Position	Labels	
<input type="checkbox"/>	Rerun rerun@gmail.com	ACTIVE	Administration	Admin QA	...
<input type="checkbox"/>	Skylar Hays skylarhays@gmail.com	ACTIVE	QA Lead	QA	...
<input type="checkbox"/>	Jamir Torres jamirtorres@gmail.com	ACTIVE	Support	Support	...
<input type="checkbox"/>	Pedro Maddox pedromaddox@gmail.com	ACTIVE	SE	QA	...
<input type="checkbox"/>	Kayley Gamble KayletQ@gmail.com	SUSPENDED	PM	PM	...
<input type="checkbox"/>	Lozenro Adkins lozenro@gmail.com	SUSPENDED	Test Lead	Tester	...
<input type="checkbox"/>	Cason Alvarado rerun@gmail.com	SUSPENDED	Back end	SE	...
<input type="checkbox"/>	Lisa Decker LisaDe@gmail.com	DEACTIVE	Front end	FE	...
<input type="checkbox"/>	Fisher James fisherJ@gmail.com	DEACTIVE	Designer	Design	...

0 of 10 row(s) selected.

Rows per page10Page 1

2.4.2.4. Project overview

Pronjenti

V1.0.0

For you

Projects

Project 1

Project 2

Project 3

View all projects

Server performance

Projects > My project 1

MPJ1 board

On track

Overview

Kanban board

List

Epic

Test case

0 completed

in the last 3 days

12 updated

in the last 3 days

1 epic created

in the last 3 days

7 days

to the end of Sprint 21

Status overview

To do

35%

35

In progress

5%

35

In testing

40%

35

To do

35 items

15%

35

Done

5%

35

Priority breakdown

Highest

5

High

10

Medium

5

Low

5

Lowest

5

Recent activity

Members

Tasks progress

Khanh Huy Pham

65%

Xuan Ngoc Pham

45%

Xuan Ngoc Pham

25%

Nhan Phi

12%

Epic progress

SCRUM-252 VA_API QR Public

65%

10%

SCRUM-729 VA_Tich hpp HPAY (urgent)

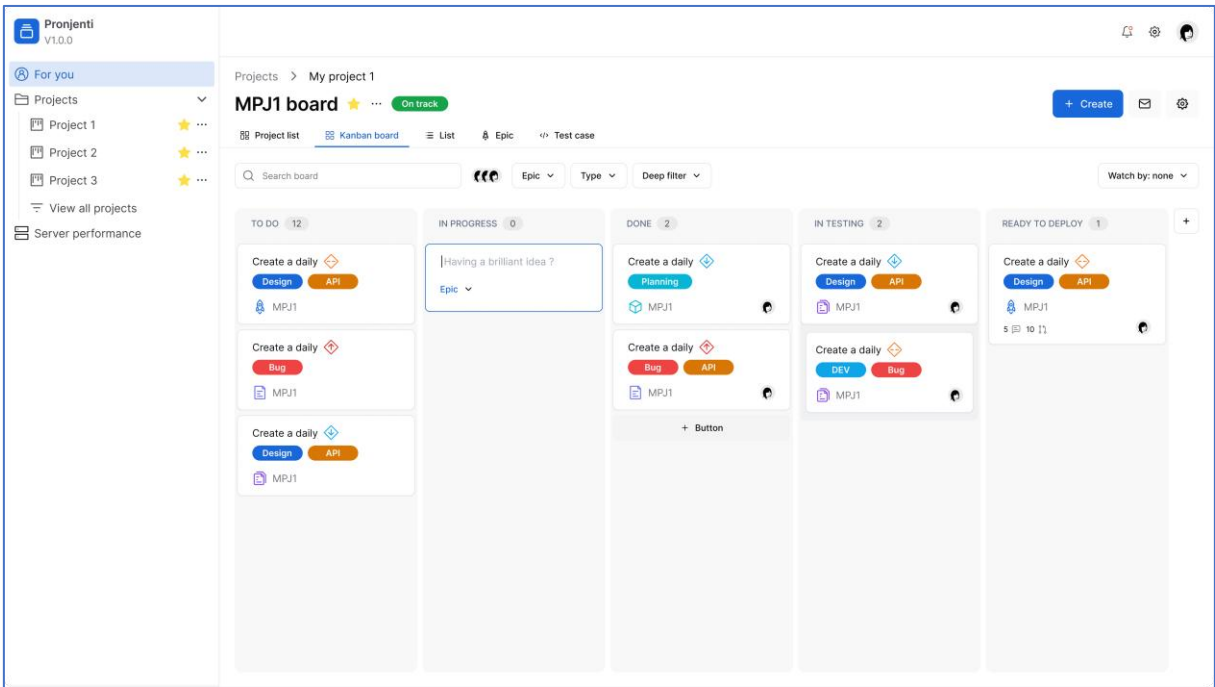
50%

45%

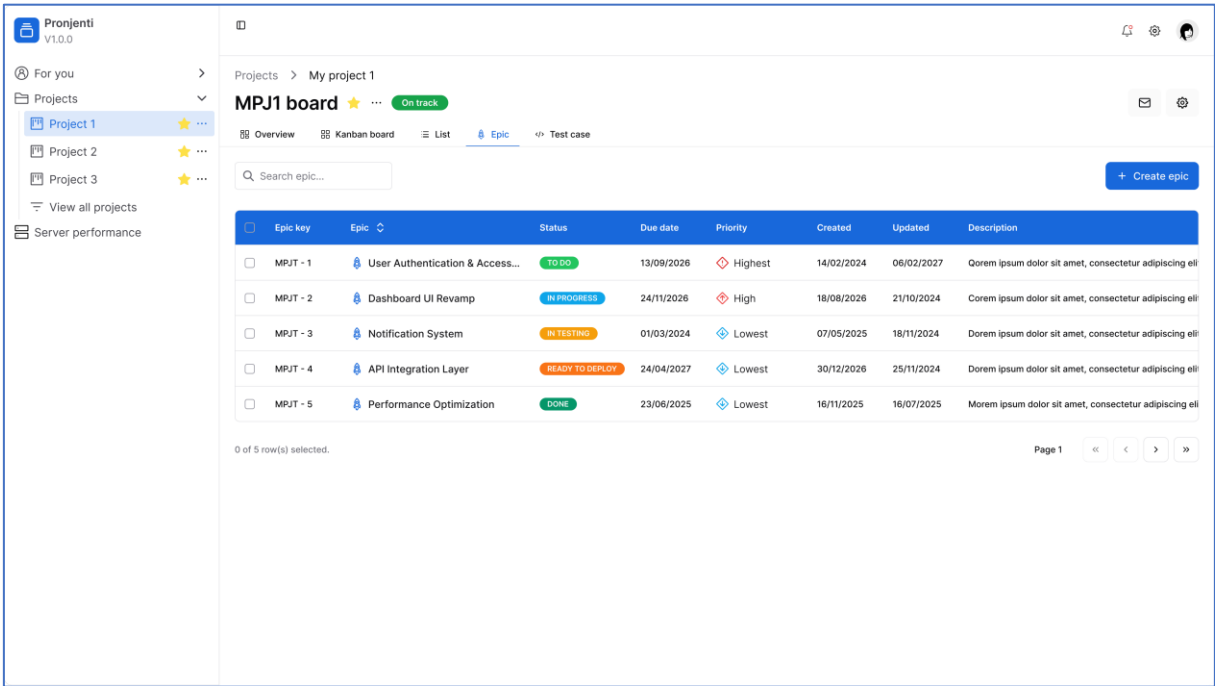
SCRUM-608 VC_Tich hpp Airwallex

26

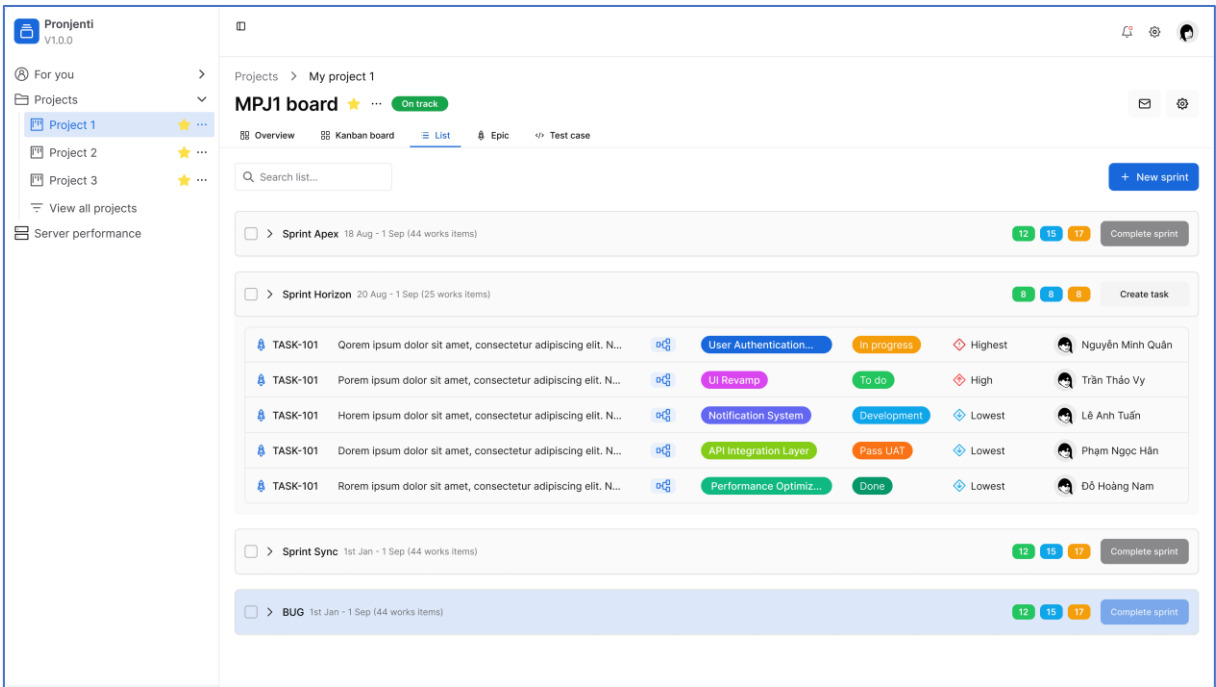
2.4.2.5. Kanban board



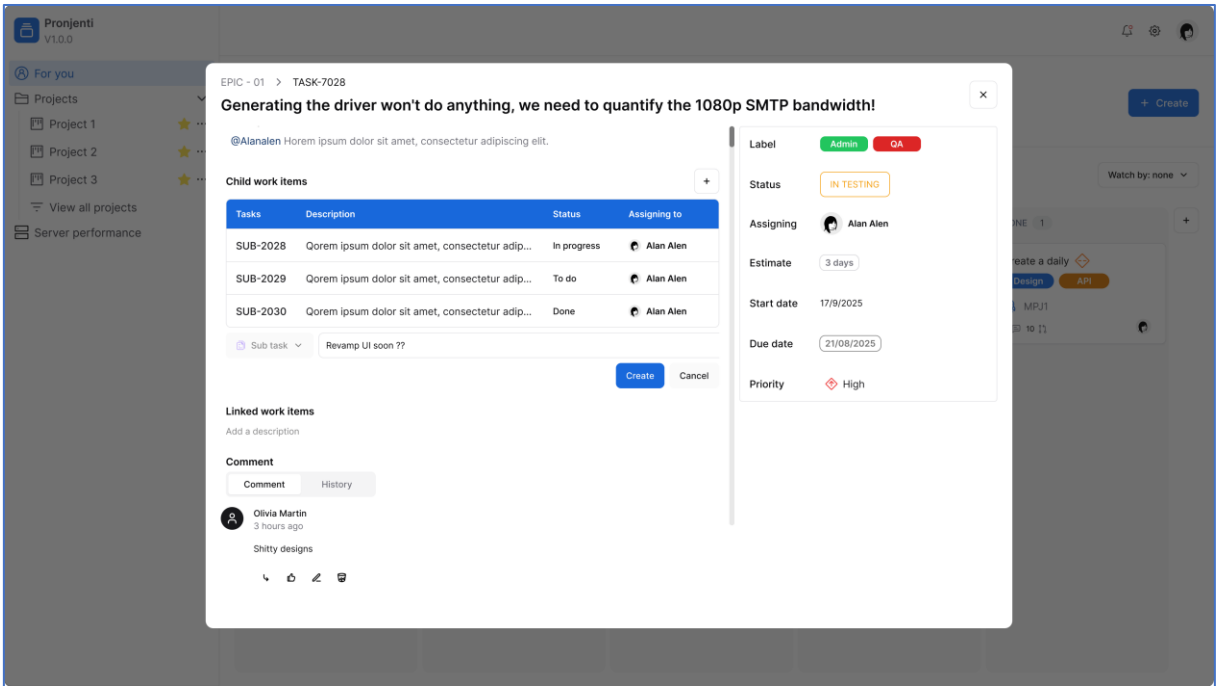
2.4.2.6. Epic list



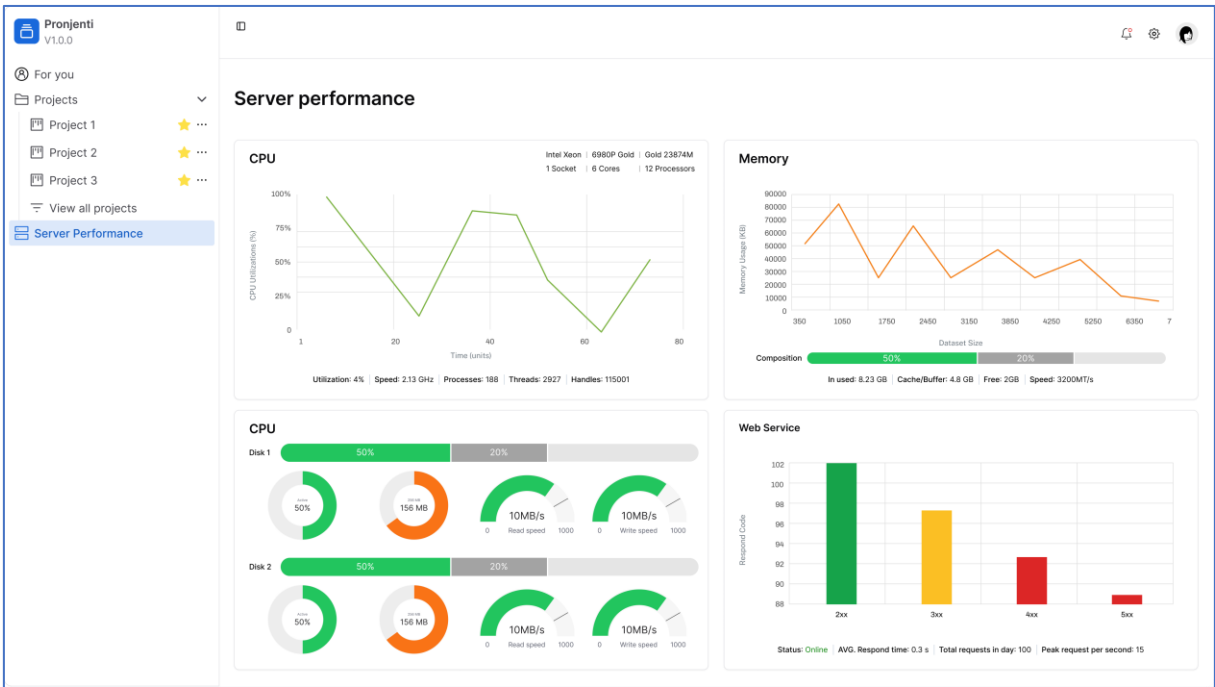
2.4.2.7. Task list



2.4.2.8. Task creation



2.4.2.9. Server performance



2.4.2.10. Testcase management

Pronjenti V1.0.0

Projects > My project 1

MPJ1 board On track Create +

Summary Kanban board **List** Test case

Search list...

+ New test case Delete Refresh

ID	Name	Execution status	Last executed	Reporter	Status	Type
TC 20	Create Author	NOT RUN	2025-10-12	Joe.P Smith	Draft	Scenario
TC 21	Create Book	NOT RUN	2026-06-05	Joe.P Smith	Draft	Scenario
TC 22	Login to Application	PASSED	2024-04-01	Joe.P Smith	Draft	Scenario
TC 23	Open Up Web Browser	NOT RUN	2023-09-14	Joe.P Smith	Ready to Test	Functional
TC 24	Delete a Option	NOT RUN	2025-02-02	Joe.P Smith	Ready to Test	Functional

0 of 5 row(s) selected. Rows per page: 5 Page 1 of 5

3. Technical Specifications

3.1. Technology stack

3.1.1. Frontend technology – ReactJS

React.js (also known as React or ReactJS) is an open-source, front-end JavaScript library developed by Meta (formerly Facebook). Its primary purpose is to facilitate the creation of interactive and dynamic user interfaces for web and mobile applications.

There are several key advantages leading to the chosen of ReactJS:

- **Component-Based Architecture:** ReactJS allows user interface to be broken down into reusable components. For Projenti, elements like a task card on the Kanban board, a task information pop-up or a sprint detail table could be built as self-contained components. This modular approach significantly speeds up development, simplifies testing and maintenance, and ensures a consistent user experience across the platform.
- **Efficient Rendering with the Virtual DOM:** Project management tools often handle large amounts of data update in a short time, especially . ReactJS uses a Virtual DOM to optimize performance by only updating the specific elements that have changed, rather than re-rendering the entire page. For Projenti users, this means interactions like dragging and dropping a task on the Kanban board or updating a project timeline will feel instantaneous and smooth, even in complex projects with hundreds of tasks.
- **Declarative UI for Simplified State Management:** ReactJS simplifies the handling of complex application states. When a user updates a task's status, the UI components that display this information—such as the Kanban board, task lists, and the main dashboard—will automatically and efficiently update to reflect the change. This reduces complexity for developers and minimizes the risk of data inconsistencies across the application.
- **Scalability and Maintainability:** Projenti is designed to grow, with future plans for advanced features. React's component-based structure is highly scalable, allowing new functionalities to be added over time without affecting existing code. This ensures that the platform can be maintained and expanded efficiently as the needs of its users evolve.

3.1.2. Backend technology – NestJs

NestJS - a Node.js framework that utilizes the Express library as its default HTTP server. This combination provides a powerful foundation for building a maintainable server-side application capable of supporting the complex requirements of a project management tool.

The value of this stack for Projenti is demonstrated below:

- **Structured and scalable architecture:** NestJS enforces a highly organized, modular architecture. Since Projenti system is broken down into distinct modules (e.g., Projects, Tasks, User Authentication), each with its own controllers for handling requests and

services for business logic. Therefore, this structure is considered to be ideal for managing complexity, making the codebase easier to maintain and scale as new features like advanced reporting or integrations are added in the future.

- **Transition to TypeScript syntax:** This framework provides static typing, which allows developers to catch potential errors during the development phase, long before they can impact production. The explicit type definitions improve code auto-completion in modern editors and make the codebase more self-documenting, which is crucial for a data-intensive application like Projehti. Ultimately, this structured approach leads to a more robust and reliable backend, significantly accelerating the entire development process by reducing debugging time and improving code quality from the start.
- **High performance and testability:** The combination of NestJS's architecture and the efficiency of the Express library ensures high performance in handling API requests, providing a fast and responsive experience for Projehti users. Furthermore, NestJS's design, which heavily utilizes Dependency Injection, makes the application highly testable. This allows for the creation of thorough unit and end-to-end tests, which is critical for ensuring the stability and correctness of the platform.

3.2. Data management

3.2.1. Database technology

MySQL is the selected open-source relational database management system that serves as the core data foundation for the Projehti platform. Renowned for its speed, reliability, and widespread adoption, MySQL is well-suited to the demands of a collaborative project management and DevOps application.

- **Data integrity and reliability:** Projehti manages parallel projects combined from multi-level components such as tasks, assignments, and test results, which demands a fast and precise data query execution. MySQL's robust support for ACID transactions ensures that every operation processed consistently and safely, even under high concurrency. This provides assurance that no data is lost or corrupted, a requirement for tracking project progress and accountability.
- **Complex relationship handling:** The data structure of Projehti provides multiple interconnected relationship not only inside project but also the role of member between them. In order to solve that, our developers harness MySQL's relational schema design and optimized join operations to make it possible to efficiently model and query these relationships. This enables Projehti to generate dynamic views such as Kanban boards, dependency reports, and detailed activity histories without performance bottlenecks.
- **High concurrency and performance:** As mentioned above, there are always distributed development teams working in parallel in Projehti. With MySQL's efficient indexing strategies and transaction handling, Projehti could sustain simultaneous read and write operations at scale, ensuring responsive data and real-time updates in every component.

4. System Development

4.1. Developing team

Role	Quantity	Personnel information
Project Sponsor	1	25+ experience in Business and Leadership Management in No 1 MNC Giant Tech Vendor.
Project Manager	1	5+ years of experience in managing software development projects with Agile/Scrum methodologies, responsible for planning, coordination, and ensuring timely delivery.
Business Analyst	1	4+ years of experience in gathering and analyzing business requirements, translating them into functional specifications. Excelling in technical implementation.
Designer	1	4+ years in UI/UX design with a focus on creating intuitive and user-friendly interfaces.
Front-end Engineer	2	4+ years of experience developing user interfaces (UI) with modern frameworks (e.g., React, Vue). Experienced in building complex interactive components such as Kanban boards and data metric dashboards.
Back-end Engineer	2	4+ years of experience building APIs and microservices. Expertise in integrating with third-party services like GitHub/GitLab and handling the core logic for features
Tester	1	3+ years of experience in software quality assurance, skilled in defining test scenarios, reporting defects, and ensuring product reliability.

4.2. Development timeline

Description	Task	Duration (weeks)
Phase 1: Project definition and setup		
	Concept definition	2
	Budget Estimate	1
	Requirement analysis	0.5
	Risk identification	0.5
	Technology definition	1

Description	Task	Duration (weeks)
	Architecture Design	1
Phase 2: Sprint 1 – Account and Project Management Function		
	Planning	0.5
	UI/UX Design	1
	Development	3.5
	Testing	0.5
	Deployment	0.5
Phase 3: Sprint 2 – Feedback Management and Monitoring Tool		
	Planning	0.5
	UI/UX Design	1
	Development/Bug fixing	2
	Testing	0.5
	Deployment	0.5
Phase 4: Sprint 3 – Third party integration		
	Planning	0.5
	UI/UX Design	1
	Development/Bug fixing	1
	Testing	0.5
	Deployment	0.5
Phase 5: Project finalization		
	Final testing	1
	Document delivery	0.5
	Product delivery	0.5
Planning		6
Development		16
Reverse time		2
Total		24

4.3. Success metric

Metric	Description	Target
Project Management Accuracy	Backlog creation, task assignment, and Kanban workflows operate exactly as specified.	100% alignment with documented requirements
Feedback Collection Accuracy	Feedback forms, survey submissions, work correctly and capture valid responses.	100% alignment with documented requirements
Monitoring Dashboard Accuracy	Test results, build status, and performance metrics are displayed consistently with input data.	No mismatched or missing data points
Test Scenario & Status Update	Test scenarios can be created, updated (Pass/Fail/Blocked), and reflected in system status.	100% alignment with documented requirements
Repository Integration	OAuth connection and repository imports (GitHub/GitLab/Bitbucket) run reliably, with commit history and CI/CD status shown.	Integration validated successfully
Error-Free Operation	System functions execute without critical or blocking errors during acceptance testing.	Zero critical defects at handover
Functional Completeness	All features listed in the MVP scope are implemented and delivered.	Full coverage of MVP feature set

5. Conclusions & Future Directions

5.1. Enhancement Plans

5.1.1. Maintenance Plan

Projenti will be supported by a proactive maintenance plan to guarantee stability, security, and continuous improvement. The system will undergo regular updates that include bug fixes, security patches, and performance optimization to ensure consistent reliability. Real-time monitoring tools will be deployed to track server health, error rates, and response times, allowing the team to respond swiftly to anomalies. In addition, periodic security and compliance audits will be conducted to maintain industry standards and protect user data. This structured approach minimizes downtime while ensuring that the platform remains dependable and scalable as usage grows.

5.1.2. Customer Support

Projenti will provide assistance through multiple channels, including email, live chat, The support team will operate under a tiered Service-Level Agreement (SLA), ensuring fast response times for both standard and critical incidents. To complement direct support, users will also have access to a comprehensive knowledge base, FAQs, and video tutorials that encourage self-service. For enterprise clients, dedicated customer success managers will be assigned to facilitate onboarding, monitor adoption, and provide customized guidance throughout the customer journey.

5.1.3. Future Functions

5.2. Conclusions

Projenti is positioned as more than just a DevOps platform; it is a strategic enabler for digital transformation and continuous improvement in software delivery. By consolidating project management, testing, monitoring, and feedback into a unified environment, it addresses critical challenges such as long cycle times, fragmented toolchains, and lack of transparency. The defined success metrics provide a measurable framework for evaluating its impact, while the enhancement plans ensure that the platform will remain reliable, user-friendly, and adaptable to evolving requirements. With a forward-looking roadmap that includes AI-driven testing, advanced analytics, and expanded integrations, Projenti is prepared to support organizations in achieving agility, efficiency, and innovation in their development processes.