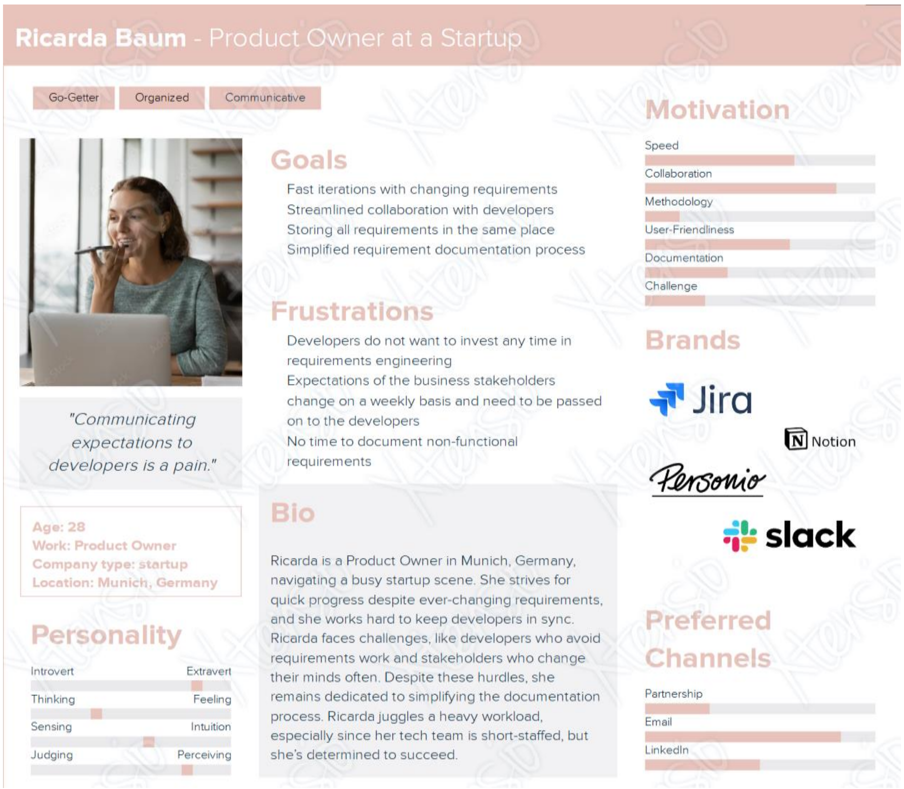
Customer Journey: ReAlign

In Munich, Germany, **Ricarda Baum**, a Product Owner in a startup, is responsible for leading project management and maintaining clarity on project objectives. However, in her recent experiences, unclear, changing, and inconsistent requirements led to project delays and frustrations for her team. Similarly, **Alex Johnson**, an experienced software developer in a big company, also faces challenges in implementing satisfying software because of the problems with project requirements. Recently, Ricarda and Alex agreed to collaborate on a project. They decided to prioritize requirement engineering to achieve better results. In a serendipitous turn of events, Ricarda's friend introduced her to **ReAlign,** a promising new platform for requirements engineering. Intrigued by the potential to simplify their workflow, enhance collaboration, and avoid frustrations in collaboration, Ricarda and Alex decided to give ReAlign a try for their latest project.



Ricarda Baum - Product Owner at a Startup

Go-Getter | Organized | Communicative

"Communicating expectations to developers is a pain."

Age: 28
Work: Product Owner
Company type: startup
Location: Munich, Germany

Goals

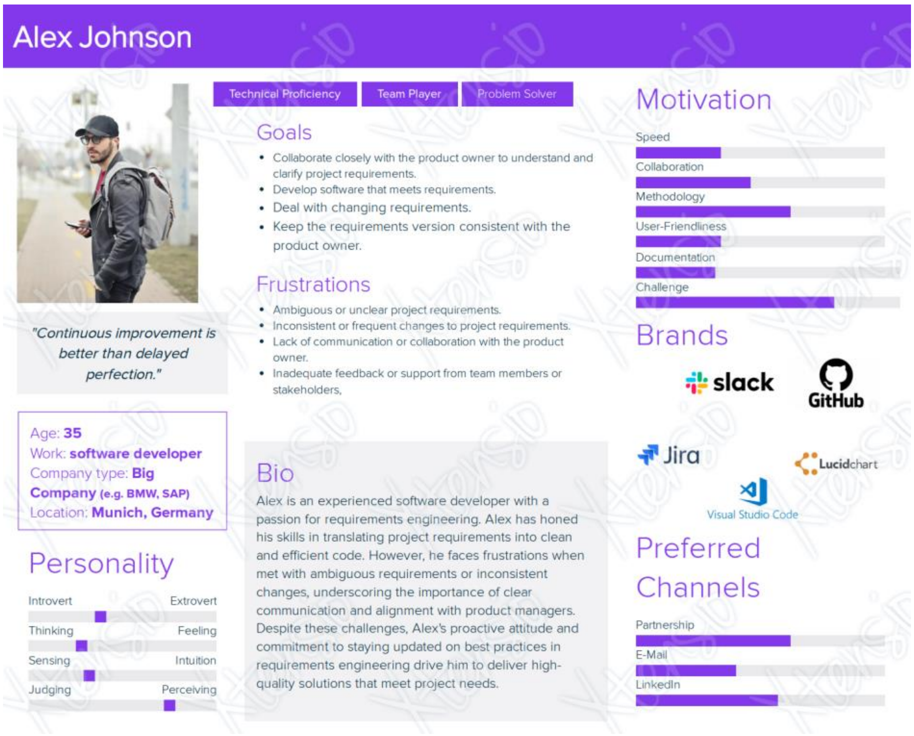
Fast iterations with changing requirements
Streamlined collaboration with developers
Storing all requirements in the same place
Simplified requirement documentation process

Frustrations

Developers do not want to invest any time in requirements engineering
Expectations of the business stakeholders change on a weekly basis and need to be passed on to the developers
No time to document non-functional requirements

Bio

Ricarda is a Product Owner in Munich, Germany, navigating a busy startup scene. She strives for quick progress despite ever-changing requirements, and she works hard to keep developers in sync. Ricarda faces challenges, like developers who avoid requirements work and stakeholders who change their minds often. Despite these hurdles, she remains dedicated to simplifying the documentation process. Ricarda juggles a heavy workload, especially since her tech team is short-staffed, but she's determined to succeed.



Alex Johnson

Technical Proficiency | Team Player | Problem Solver

"Continuous improvement is better than delayed perfection."

Age: **35**
Work: **software developer**
Company type: **Big Company (e.g. BMW, SAP)**
Location: **Munich, Germany**

Goals

- Collaborate closely with the product owner to understand and clarify project requirements.
- Develop software that meets requirements.
- Deal with changing requirements.
- Keep the requirements version consistent with the product owner.

Frustrations

- Ambiguous or unclear project requirements.
- Inconsistent or frequent changes to project requirements.
- Lack of communication or collaboration with the product owner.
- Inadequate feedback or support from team members or stakeholders,

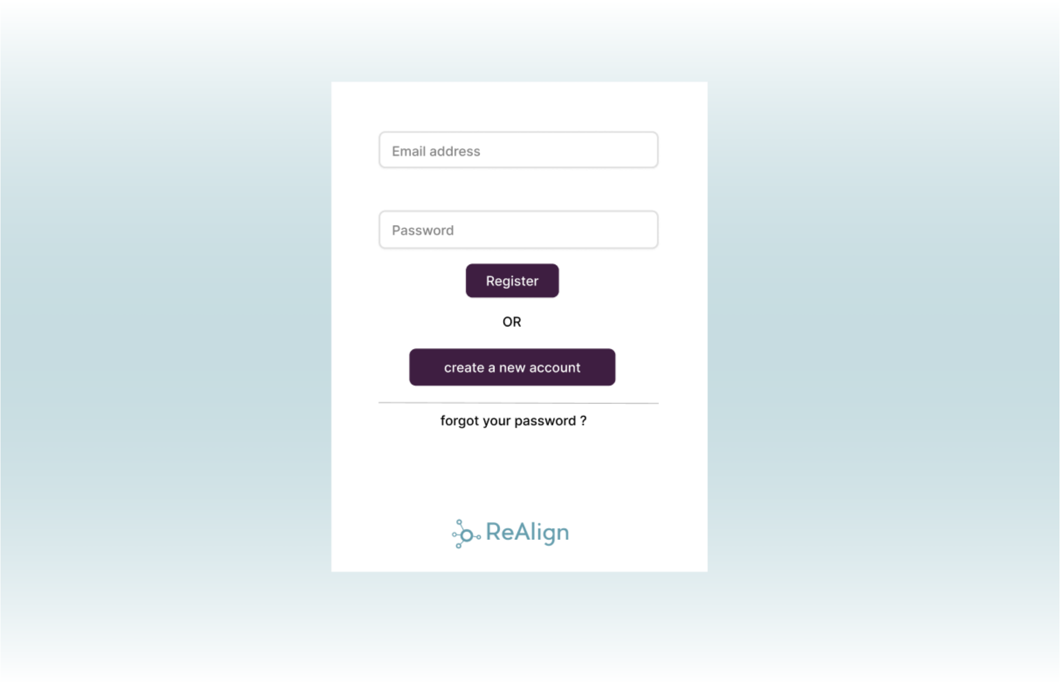
Bio

Alex is an experienced software developer with a passion for requirements engineering. Alex has honed his skills in translating project requirements into clean and efficient code. However, he faces frustrations when met with ambiguous requirements or inconsistent changes, underscoring the importance of clear communication and alignment with product managers. Despite these challenges, Alex's proactive attitude and commitment to staying updated on best practices in requirements engineering drive him to deliver high-quality solutions that meet project needs.

The first step in their journey begins with **registration and login.** Ricarda visits the ReAlign website and is greeted by an interface that allows her to sign up. With a few clicks, she navigates to the "Sign Up" button and proceeds to fill out the registration form with her details such as name, surname, and so on. Upon submission of the form, a confirmation email is sent to Ricarda's provided email address. Ricarda verifies her account by clicking the email link and ReAlign's backend securely stores her registration information in the database. Ricarda's account registration is completed and created in the database. When Ricarda logs in to ReAlign using her credentials, the platform manages the authentication process. Her login details are verified against the platform's database, ensuring secure access. Upon first logging in, Ricarda is presented with the option to choose between a free trial or subscription mode. Since it is Ricarda's first time using this platform, she decided to try the free trial mode first.

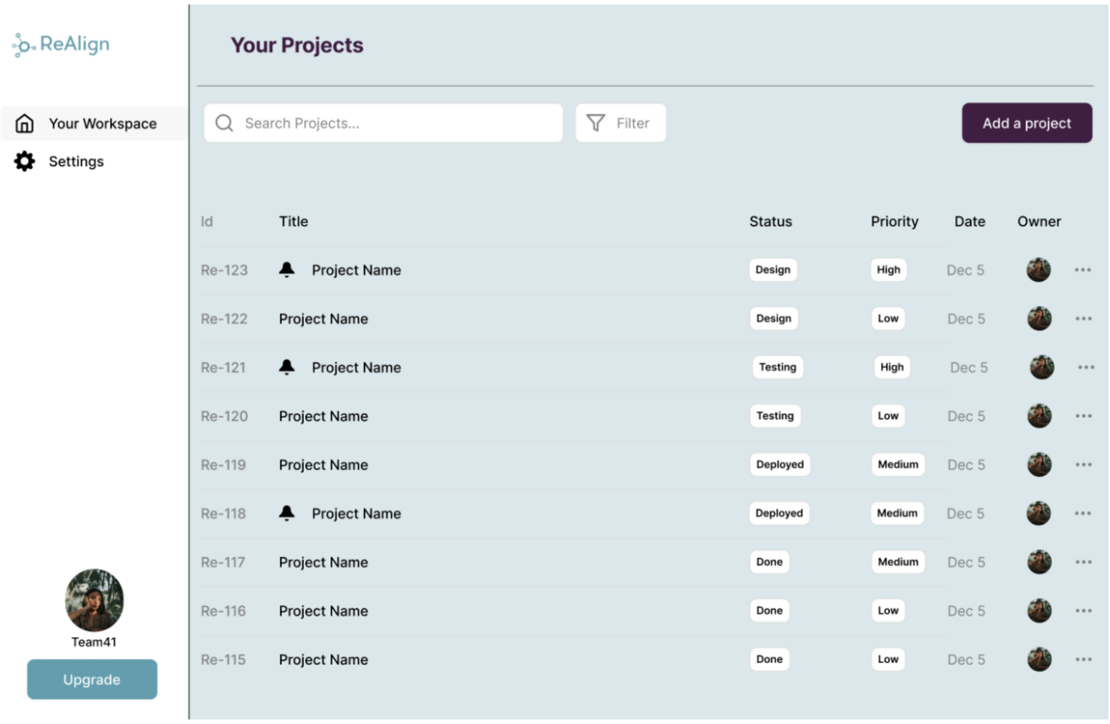
For added convenience, ReAlign offers a "Forgot password?" feature which updates the user's password in the database upon verification, ensuring that users can easily reset their passwords in case they forget them.

Alex also finds the login process seamless and was pleased to have the option of a free trial or subscription. Both Ricarda and Alex are impressed by the platform's user-friendly interface and intuitive design. They are increasingly convinced that ReAlign would be the ideal tool for their project.



Registration/Log-in Page

After successfully logging in, Ricarda is redirected to the **ReAlign dashboard**. In the dashboard, users' project data is retrieved from the database, including project names, statuses, priorities, owners, and creation dates. Ricarda finds an immediate overview of all ongoing projects, displayed in a visually appealing manner that allows her to quickly grasp the status and priority of each project at a glance. Other important project details, such as the project owner and creation date, are also displayed alongside each project. Ricarda has the option to double-click on a project name, initiating a "Read" operation that fetches project data directly from the database. This action seamlessly navigates Ricarda into the selected project. Additionally, Ricarda notices a notifications section displayed, ensuring that she stays updated on any important updates or changes within her projects. Impressed by the dashboard's efficiency, Ricarda decides to explore further. The "Upgrade" button located in the bottom left corner enables Ricarda to plan a subscription. She also notices the button to add a project, positioned in the right-upper corner of the dashboard. Furthermore, Ricarda observes the search and filter functionality available within the dashboard. The “Search and Filter” functionality involves a “Read” operation in the backend. When Ricarda uses this feature to search for specific projects or filter them based on criteria such as “high priority” or “design status,” the backend retrieves relevant project data from the database based on Ricarda’s input. This data is then returned to Ricarda, presenting her with a filtered list of projects that match her search or filter criteria, thereby enabling her to quickly find and access the projects she needs.



Id	Title	Status	Priority	Date	Owner
Re-123	🔔 Project Name	Design	High	Dec 5	•••
Re-122	Project Name	Design	Low	Dec 5	•••
Re-121	🔔 Project Name	Testing	High	Dec 5	•••
Re-120	Project Name	Testing	Low	Dec 5	•••
Re-119	Project Name	Deployed	Medium	Dec 5	•••
Re-118	🔔 Project Name	Deployed	Medium	Dec 5	•••
Re-117	Project Name	Done	Medium	Dec 5	•••
Re-116	Project Name	Done	Low	Dec 5	•••
Re-115	Project Name	Done	Low	Dec 5	•••

Dashboard/Homepage

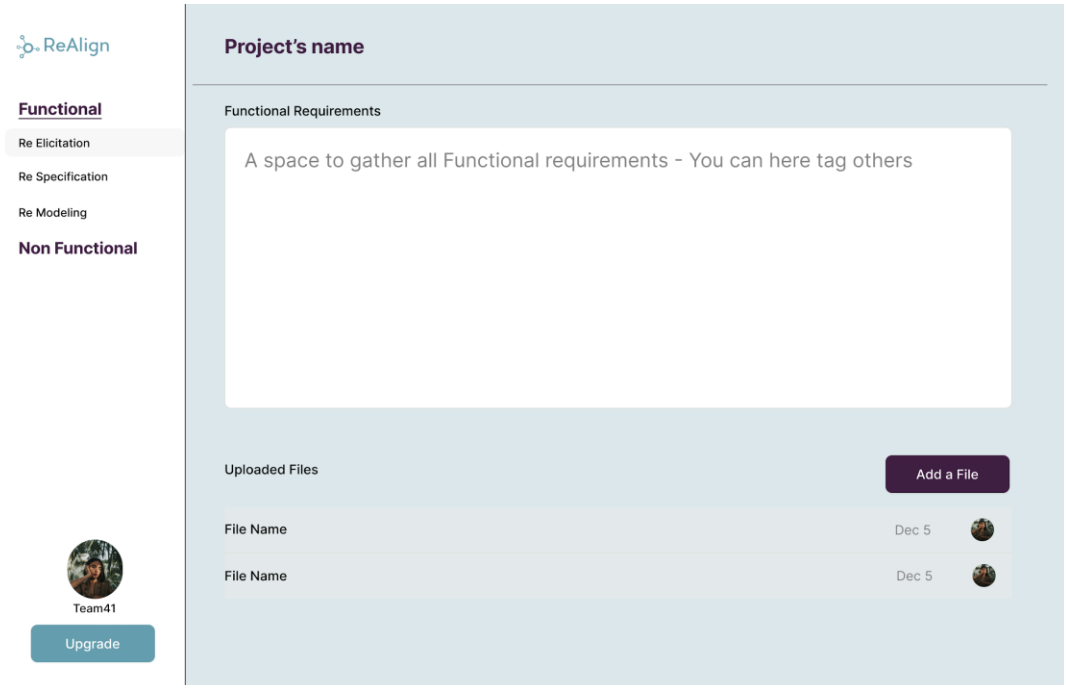
With the dashboard providing a comprehensive overview of the projects and offering convenient access to key features, Ricarda looks forward to **creating her project** which collaborates with Alex. Ricarda clicks on the "Create New Project" button on the dashboard. She swiftly fills out the project details such as name, description, deadline, and status, ensuring clarity on the project's objectives. She notices that she can set access permissions for team members, choosing from options such as Owner, Editor, or Reader. This feature allows her to tailor access levels based on each team member's role and responsibilities. Ricarda then invites Alex to join the project, setting his role as an editor to grant him permission to edit the content. With a click on the "Create Project" button, their collaborative project officially begins on ReAlign.

The backend processes the request and creates a new entry in the database for the project, storing details of the project name, description, deadline, and status. Alex receives the invitation, and he clicks on the new project listed on his dashboard. This initiates a "Read" operation in the backend and retrieves the project data from the database after role verification. As a result, Alex gains access to the project workspace, enabling him to collaborate with Ricarda effectively.

With the project successfully created, Ricarda navigates to the project workspace on ReAlign to begin working on the requirements. Within the intuitive interface, they have the option to choose between functional or non-functional requirements. Ricarda and Alex can input detailed requirements with

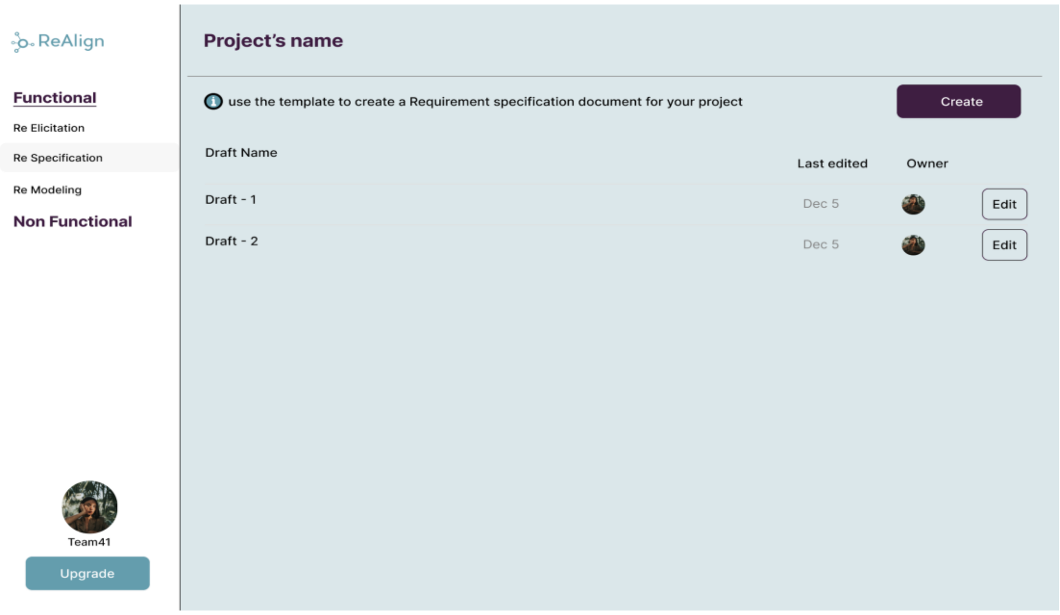
ReAlign’s features. The backend processes their inputs and creates new entries in the database for each requirement, storing details such as the requirement title, description, type, and so on. To enhance the documentation process, they leverage the platform's capabilities to attach files or link related documents directly to the requirements, providing additional context and resources. As Ricarda and Alex collaborate on refining and updating the project's requirements, the backend promptly updates the corresponding database entries, ensuring that all team members have access to the latest project information and fostering efficient communication and alignment throughout the project lifecycle.

Ricarda and Alex can **choose between different stages for functional requirements**: Requirement Elicitation, Requirement Specification, and Requirement Modelling. Under the “Functional” section, Alex clicks on **“Re Elicitation”** and starts to gather required requirements in a text area upon saving it the backend processes the request and then it is stored as a file in the database with the body, last edited by and created by. Ricarda already had a file that documented elicitation, so, Ricarda presses the “Add a File” button and proceeds to add a file from her local computer, the upload request is handled by the backend and stores the file to the respective requirement. After a successful upload, the webpage is refreshed, and Ricarda can view the file she uploaded in the same section.



Functional: Re Elicitation

After transitioning to the **"Re Specification"** stage in ReAlign, Ricarda gains access to a comprehensive overview of available documents. Each document is listed with details such as its name, owner, and last edited date. Additionally, an action button labeled "Edit" allows Ricarda to make modifications to the documents. With a clear understanding of her project needs, Ricarda decides to create a new file for specification. Upon clicking the "Create" button, she is presented with a predefined template to streamline the process. Ricarda proceeds to provide a name for the file and enters relevant information into the template. Once satisfied with the content, she saves the file. Behind the scenes, ReAlign's database seamlessly handles Ricarda's request by creating a new file and inserting the provided details such as the file name, owner, and last edited timestamp. This ensures that the newly created file is accurately recorded and readily accessible for future reference.



Project’s name

use the template to create a Requirement specification document for your project

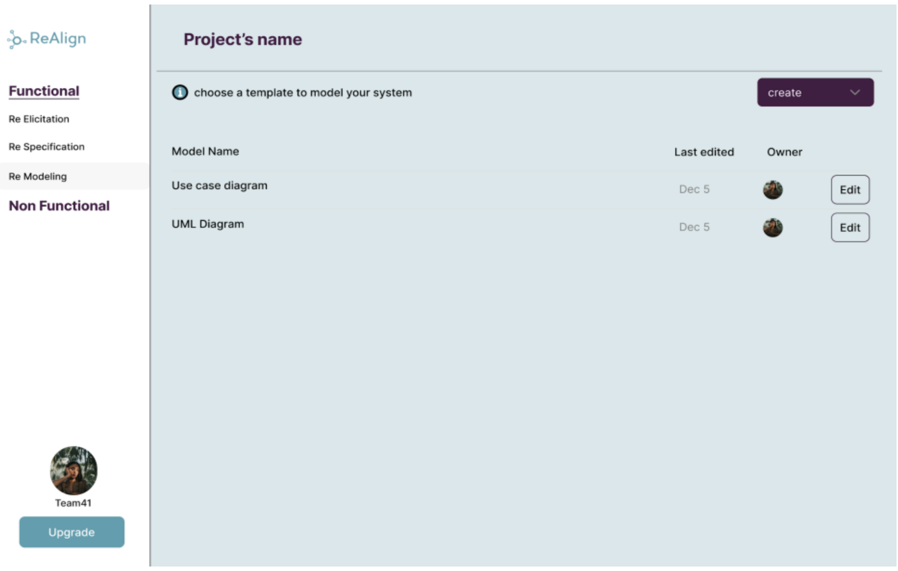
Draft Name	Last edited	Owner	
Draft - 1	Dec 5		Edit
Draft - 2	Dec 5		Edit

Re Specification

Alex who is also working on the project wants to update the specification document with new information that he is working on, so, he chooses the file that Ricarda previously created and clicks on the “Edit” button, these initiates read operation to the backend, where the file id is used to retrieve the file. After a successful read operation, Alex views the document, updates it with new information, and saves it where the backend will update the file in the database.

Ricarda then proceeds to the next and final stage of functional requirement which is “Re Modeling”, she wants to create a diagram for the requirement of her project. She clicks on “Create” button and promptly enters the model name with a type. Upon receiving the request to create a new model, ReAlign presents a workspace where Ricarda can create her UML diagram. She starts to work on the use diagram by creating actors and systems and interaction between them. Ricarda feels satisfied with the diagram she made and decides to save the model.

The backend processes this request by initiating a create data service where model name, type, and file are saved to the database. Alex wishes to view the use case diagram Ricarda prepared, so, he clicks on the model’s name, the backend processes this as a read service for the model and then displays it to Alex.



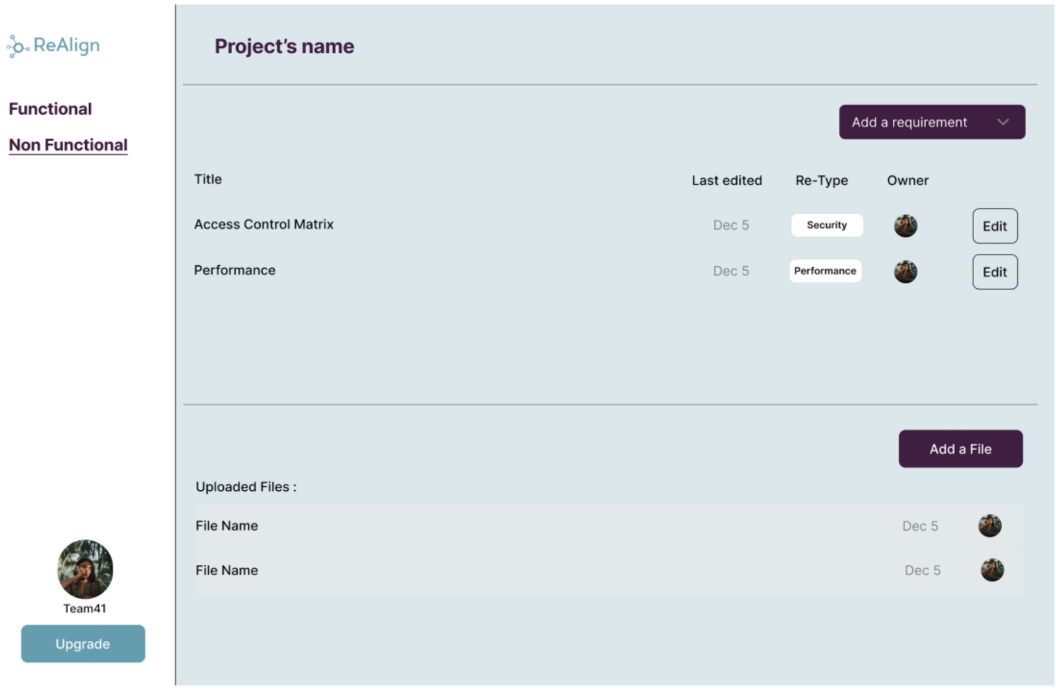
Project’s name

choose a template to model your system

Model Name	Last edited	Owner	
Use case diagram	Dec 5		Edit
UML Diagram	Dec 5		Edit

Re Modeling

After the successful creation of the model, Ricarda **visits the "Non-Functional" requirement** section to add a security measure for her requirement. Upon clicking the "Non-Functional" section, Ricarda gets the overall list view of nonfunctional requirements, an "Add a requirement" button to add a new NFR, and an "Add a File" button to upload external attachments. Ricarda clicks on the "Add a requirement" button fills in with the NFR title, security as a type, and saves it. Ricarda then adds an access control matrix as a security model used to define and manage permissions within a system or application to the requirement, this request is processed as a create service then the requirement is saved to the database with its attribute title, type, name, createdBy, createdAt with respective projectId. After successful insertion, Ricarda can see the NFR she created previously.



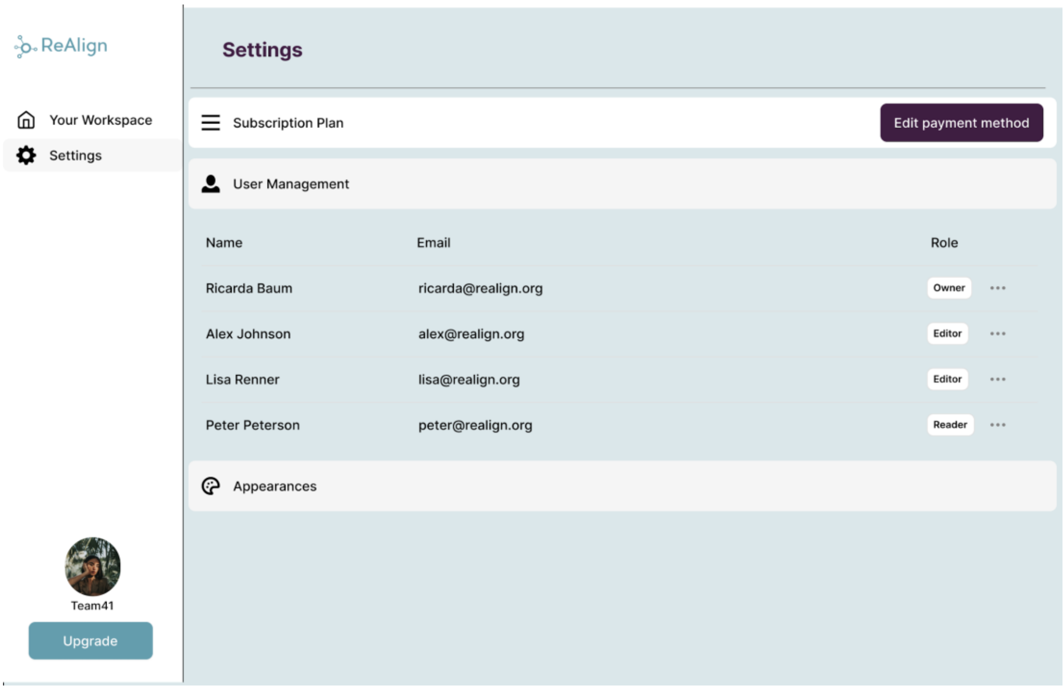
Title	Last edited	Re-Type	Owner	
Access Control Matrix	Dec 5	Security		Edit
Performance	Dec 5	Performance		Edit

Uploaded Files :

File Name	Dec 5	
File Name	Dec 5	

Non-Functional requirements

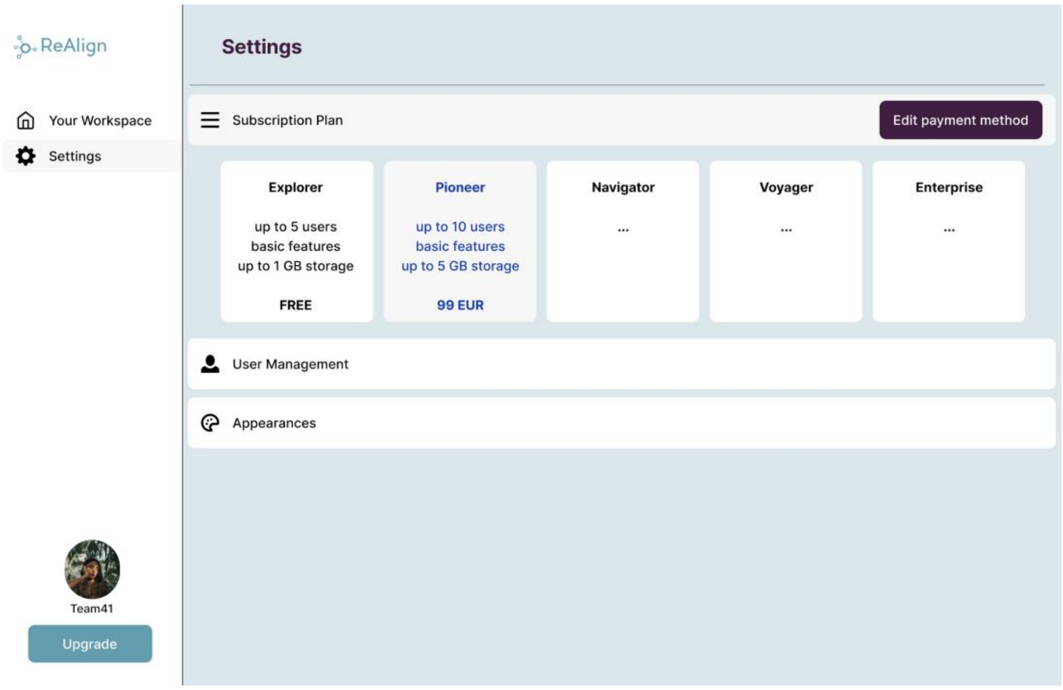
After creating a new Non-Functional Requirement (NFR), Ricarda decides to explore **the settings page** of ReAlign, a software platform she's using. Upon entering the settings section, she encounters a comprehensive overview of the subscription plan, user management, and appearance settings. Intrigued by the user management functionality, Ricarda navigates to the "User Management" section. In this section, the backend system initiates a read service to fetch all users associated with the project. As the backend processes the request, Ricarda is presented with a front-end interface displaying a list of users. Each user entry includes their name, email address, and role within the project. This detailed presentation allows Ricarda to easily identify and manage the users involved in the project. The seamless integration between the backend and frontend systems enables Ricarda to efficiently view and interact with the user data.



Name	Email	Role
Ricarda Baum	ricarda@realign.org	Owner
Alex Johnson	alex@realign.org	Editor
Lisa Renner	lisa@realign.org	Editor
Peter Peterson	peter@realign.org	Reader

User Management

After evaluating ReAlign as a requirement engineering platform for her project, Ricarda is determined to continue using its services. Considering her project's needs and future growth, she opts to purchase the "Pioneer" package, which offers support for up to 10 users and provides storage of up to 5GB. With her decision made, Ricarda proceeds to update her payment details by clicking on the "Edit payment method" button. This action allows her to securely input and manage her payment information, ensuring a seamless and hassle-free transaction process. By selecting a suitable package and updating her payment method, Ricarda demonstrates her commitment to leveraging ReAlign's features and capabilities to support her project's requirements effectively.



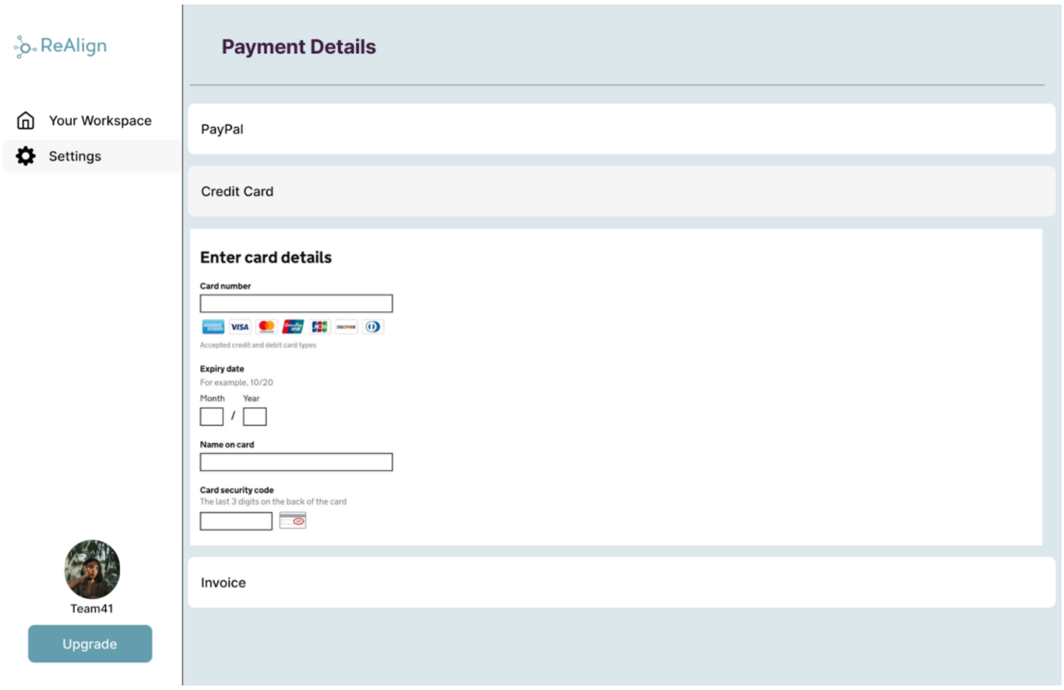
Explorer	Pioneer	Navigator	Voyager	Enterprise
up to 5 users basic features up to 1 GB storage	up to 10 users basic features up to 5 GB storage	...	...	...
FREE	99 EUR			

Subscription plan

After selecting the "Credit Card" option for payment on ReAlign, Ricarda is prompted to enter her card details. With efficiency and accuracy, she swiftly fills in the required information, including her card number, expiry date, name on the card, and security code. Upon completion, she clicks on the "Update" button to confirm and proceed with the payment process.

Behind the scenes, the backend system of ReAlign processes Ricarda's request as an update service for payment. It meticulously validates and encrypts the provided card details for security purposes. Subsequently, the backend system adds the newly provided fields, such as the card number, expiry date, name on card, and security code, to the payment database. This ensures that Ricarda's payment information is securely stored and readily accessible for future transactions.

The seamless interaction between the frontend interface and backend systems of ReAlign allows Ricarda to effortlessly manage her payment details. By promptly updating her payment information, Ricarda ensures a smooth and uninterrupted continuation of her subscription, reaffirming her commitment to utilizing ReAlign as her chosen requirement engineering platform for her project.



Payment Details

PayPal

Credit Card

Enter card details

Card number

Accepted credit and debit card types

Expiry date

For example, 10/20

Month / Year

Name on card

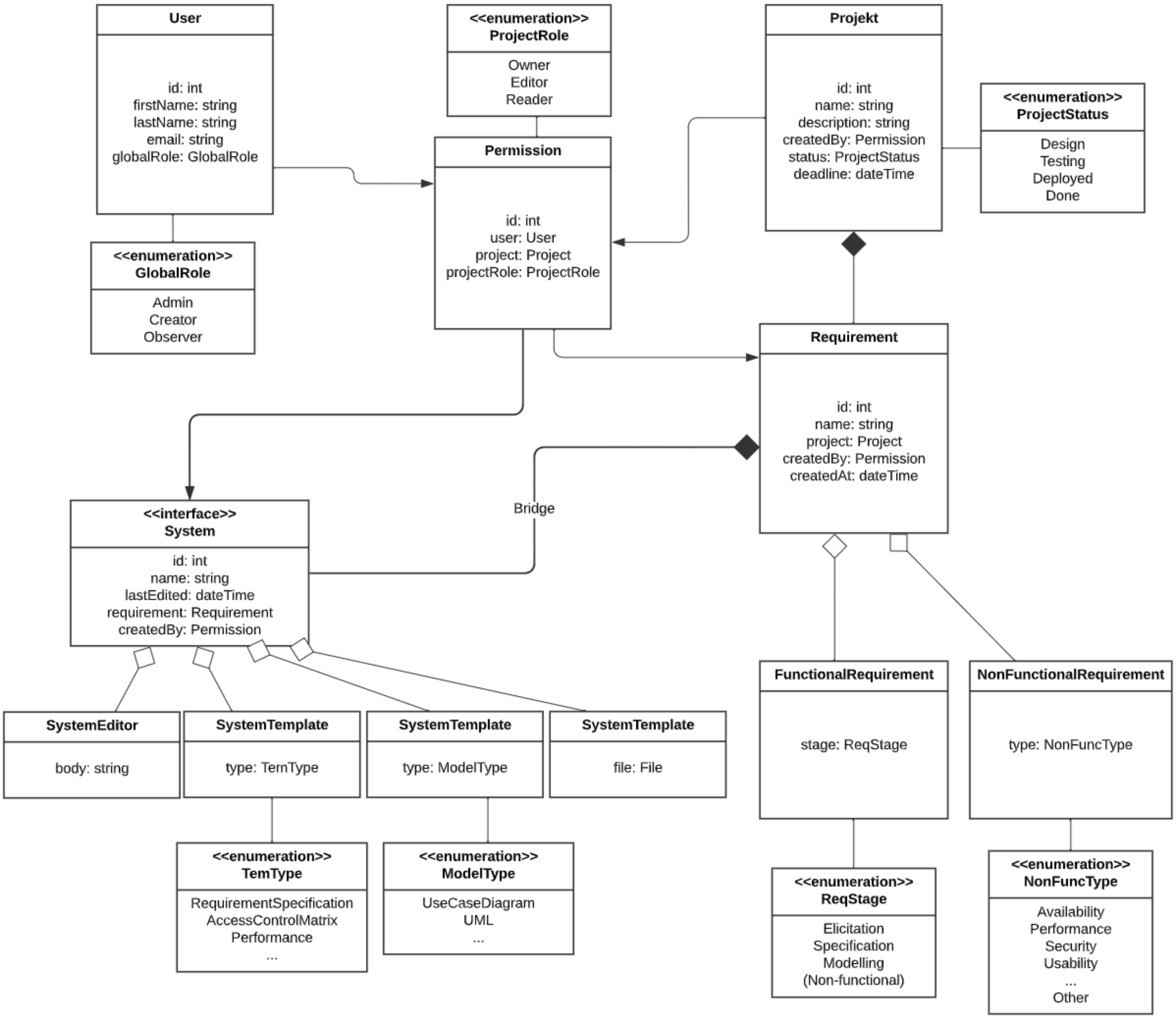
Card security code

The last 3 digits on the back of the card

Invoice

Payment integration

After successful payment by Ricarda, she can invite up to 10 users, use basic features of the system, and use up to 5 GB of storage. As Ricarda completes her transaction and updates her payment details on ReAlign, she solidifies her commitment to leveraging the platform as her primary requirement engineering solution for her project. With seamless user experience and efficient backend processing, ReAlign continues to empower Ricarda in her project management endeavors. As she moves forward with confidence, she is assured of the platform's reliability and support in achieving her project goals. Ricarda's journey with ReAlign exemplifies the platform's dedication to providing a user-centric experience and enabling seamless collaboration for project success.


User
id: int
firstName: string
lastName: string
email: string
globalRole: GlobalRole
<<enumeration>>
GlobalRole
Admin
Creator
Observer
<<enumeration>>
ProjectRole
Owner
Editor
Reader
Permission
id: int
user: User
project: Project
projectRole: ProjectRole
Projekt
id: int
name: string
description: string
createdBy: Permission
status: ProjectStatus
deadline: dateTime
<<enumeration>>
ProjectStatus
Design
Testing
Deployed
Done
Requirement
id: int
name: string
project: Project
createdBy: Permission
createdAt: dateTime
Bridge
<<interface>>
System
id: int
name: string
lastEdited: dateTime
requirement: Requirement
createdBy: Permission
SystemEditor
body: string
SystemTemplate
type: TernType
SystemTemplate
type: ModelType
SystemTemplate
file: File
<<enumeration>>
TemType
RequirementSpecification
AccessControlMatrix
Performance
...
<<enumeration>>
ModelType
UseCaseDiagram
UML
...
FunctionalRequirement
stage: ReqStage
NonFunctionalRequirement
type: NonFuncType
<<enumeration>>
ReqStage
Elicitation
Specification
Modelling
(Non-functional)
<<enumeration>>
NonFuncType
Availability
Performance
Security
Usability
...
Other